# Легенда

Решено сделать трехэтапную проверку правильности ответов (когда вы согласны с ответом - выделяйте его соответствующим маркером, пожалуйста).

* 1 человек выбрал этот ответ
* 2 человека выбрали этот ответ
* 3 человека выбрали этот ответ

Если вы не согласны с уже выбранным ответом, то выделите тот, который кажется правильным настороженно-оранжевым цветом, а номер вопроса отметьте осторожно-красным.

Ответ из исходного файла отмечен небесно-голубым (не проверен).

Все цвета сохранены в палитре в отделе “другой” (настороженно-оранжевый - второй, небесно-голубой - последний).

# 

# 

# Вопросы с теста

# **Какие предопределённые коммуникаторы есть в MPI при запуске программы?**

# MPI\_COMM\_NULL

# никаких

# MPI\_COMM\_SEND

# MPI\_COMM\_PROC

# MPI\_COMM\_WORLD

# MPI\_COMM\_SELF

# **В чём особенность закона Амдала применительно к компьютерам с общей памятью:**

# последовательная часть программы увеличивается за счёт синхронизации доступа к общим данным

# верного ответа нет

# очень большое число процессоров приводит к задержкам на коммуникации

# для компьютеров с общей памятью закон Амдала перестаёт действовать

# База вопросов

## Часть 1. Параллелизм и ускорения

### Конвейеры

Длина конвейера L - число его ступеней. Время обработки входа длины N равно

Если i-я ступень конвейера обрабатывает данные за T\_i, то после разгона конвейер будет выдавать результаты с периодом max(T\_i). Время обработки в этом случае

.

За конвейер успевает заполниться и выдать первый элемент, оставшиеся (N - 1) выдаются затем с периодом max(T\_i).

Совет от Соника.

Пока обнаружены задачи двух типов: выбрать максимум из T\_i и посчитать T по самой большой формуле.

Вопрос: В конвейерном устройстве есть 5 ступеней, срабатывающих за 2, 1, 5, 4 и 3 единицы времени соответственно. При обработке массива данных (в режиме максимальной загрузки конвейера) это устройство будет выдавать результат:

* каждую 5-ю единицу времени
* каждую 3-ю единицу времени
* каждую 2-ю единицу времени
* каждую единицу времени
* верного ответа нет
* каждую (2+1+5+4+3=15)-ю единицу времени

Вопрос. В конвейерном устройстве есть 5 ступеней, срабатывающих за 2, 1, 4, 4 и 3 единицы времени соответственно. При обработке массива данных (в режиме максимальной загрузки конвейера) это устройство будет выдавать результат:

* каждую (2+1+4+4+3=14)-ю единицу времени
* каждую 2-ю единицу времени
* каждую 3-ю единицу времени
* каждую 5-ю единицу времени
* верного ответа нет
* каждую единицу времени
* каждую 4-ю единицу времени

Вопрос. В конвейерном устройстве есть 5 ступеней, срабатывающих за 2, 1, 5, 4 и 3 единицы времени

соответственно. При обработке массива данных (в режиме максимальной загрузки конвейера) это устройство будет выдавать результат:

* каждую 5-ю единицу времени
* каждую 3-ю единицу времени
* каждую 2-ю единицу времени
* каждую единицу времени
* верного ответа нет
* каждую (2+1+5+4+3=15)-ю единицу времени

Вопрос. В конвейерном устройстве есть 5 ступеней, срабатывающих за 2, 1, 4, 1 и 3 тактов соответственно. При обработке массива данных (в режиме максимальной загрузки конвейера) это устройство будет выдавать результат:

* Каждый 5-й такт.
* Каждый 3-й такт.
* Каждый 4-й такт.
* Верного ответа нет.
* Каждый (2+1+4+1+3=11)-й такт.
* Каждый 2-й такт.
* Каждый такт.

Вопрос. В конвейерном устройстве есть 4 ступени, срабатывающих за 2, 1, 4 и 3 тактов соответственно. При обработке массива данных (в режиме максимальной загрузки конвейера) это устройство будет выдавать результат:

* Верного ответа нет.
* Каждый такт.
* Каждый 2-й такт.
* Каждый 5-й такт.
* Каждый 4-й такт.
* Каждый (2+1+4+3=10)-й такт.
* Каждый 3-й такт.

Вопрос. В конвейерном устройстве есть 5 ступеней, срабатывающих за 3, 1, 2, 1 и 3 тактов соответственно. При обработке массива данных (в режиме максимальной загрузки конвейера) это устройство будет выдавать результат:

* Каждый 2-й такт.
* Каждый 3-й такт.
* Каждый 5-й такт.
* Каждый 4-й такт.
* Каждый такт.
* Каждый (3+1+2+1+3=10)-й такт.
* Верного ответа нет.

Вопрос. В конвейерном устройстве есть 5 ступеней, срабатывающих за 3, 1, 2, 4 и 3 тактов соответственно. При обработке массива данных (в режиме максимальной загрузки конвейера) это устройство будет выдавать результат:

* Верного ответа нет.
* Каждый 5-й такт.
* Каждый такт.
* Каждый (3+1+2+4+3=13)-й такт.
* Каждый 4-й такт.
* Каждый 2-й такт.
* Каждый 3-й такт.

Вопрос. В конвейерном устройстве есть 5 ступеней, срабатывающих за 3, 1, 2, 1 и 3 тактов соответственно. При обработке массива данных (в режиме максимальной загрузки конвейера) это устройство будет выдавать результат:

* Каждый 2-й такт.
* Каждый 3-й такт.
* Каждый 5-й такт.
* Каждый 4-й такт.
* Каждый такт.
* Каждый (3+1+2+1+3=10)-й такт.
* Верного ответа нет.

Вопрос. Конвейерное ФУ деления состоит из пяти ступеней, срабатывающих за 2, 4, 4, 1 и 1 такт cоответственно. Чему равно наименьшее число тактов, за которое можно обработать 40 пар аргументов на данном устройстве?

* верного ответа нет
* 168
* 51
* 1
* 480
* 172
* 12

Вопрос. В конвейерном устройстве есть 3 ступени, срабатывающих за два такта каждая. За сколько тактов это устройство обработает 6 пар аргументов?

* Верного ответа нет.
* 12
* 1
* 3
* 6
* 16
* 20

Вопрос. В конвейерном устройстве есть 4 ступени, срабатывающих за один такт каждая. За сколько тактов это устройство обработает 5 пар аргументов?

* 9
* 15
* 1
* Верного ответа нет.
* 3
* 5
* 7
* 8

Вопрос. Конвейерное ФУ деления состоит из пяти ступеней, срабатывающих за 2, 3, 5, 1 и 1 такт соответственно. Чему равно наименьшее число тактов, за которое можно обработать 40 пар аргументов на данном устройстве?

* 212
* 5
* 12
* верного ответа нет
* 207
* 40
* 200
* 1
* 480

Вопрос. Конвейерное функциональное устройство имеет 7 ступеней, каждая ступень срабатывает за 1 такт. За сколько тактов такое устройство обработает 777 аргументов в предположении, что аргументы в устройство всегда подаются без задержек?

* 783
* 111
* 7
* 771
* 700
* верного ответа нет
* 777

Вопрос. В конвейерном устройстве есть 7 ступеней, срабатывающих за одну единицу времени каждая. За сколько единиц времени это устройство обработает 7 пар аргументов?

* 14
* 13
* 1
* 3
* 7
* верного ответа нет
* 8

Вопрос. В конвейерном устройстве есть 5 ступеней, срабатывающих за один такт каждая. За сколько тактов это устройство обработает 3 пары аргументов?

* 7
* 5
* 15
* 8
* 1
* 3
* Верного ответа нет.

Вопрос. В конвейерном устройстве есть 5 ступеней, срабатывающих за один такт каждая. За сколько тактов это устройство обработает 7 пар аргументов?

* 11
* других вариантов не было

### 

### Закон Амдала

Обозначим T\_p - время работы программы на p процессорах.

Закон Амдала имеет вид

Здесь p- число процессоров, s = T\_1 / T\_p - ускорение работы программы.

f - доля последовательных операций в программе (безразмерная величина из [0, 1]).

Из формулы следует, что ускорение в 1 / f раз и более невозможно.

Если вместо неравенства рассматривать равенство, то можно выразить

Совет от Соника.

В задачах далее “ускорить (1-f)\*100% программы, чтобы ускорить всю программу в s раз”. f и s нужно просто подставить в формулу для p.

Вопрос. Во сколько раз, согласно закону Амдала, нужно ускорить 50% программы, чтобы ускорить всю программу в 2 раза?

* 1000
* 2
* 8
* 3
* Верного ответа нет.
* 100
* Ускорить невозможно.
* 4

Вопрос. Во сколько раз, согласно закону Амдала, нужно ускорить 90% программы, чтобы ускорить всю программу в 8 раз?

* 8
* 10
* 80
* 36
* 5
* 1000
* Верного ответа нет.
* Ускорить невозможно.

Вопрос. Во сколько раз нужно ускорить 90% программы, чтобы ускорить всю программу в 5 раз?

* 450
* 9
* правильного ответа нет
* 10
* невозможно
* 5
* 6
* 18

Вопрос. Во сколько раз нужно ускорить 80% программы, чтобы ускорить всю программу в 6 раз?

* правильного ответа нет
* 6
* 40/3
* 480
* невозможно
* 120
* 80

Вопрос. Во сколько раз, согласно закону Амдала, нужно ускорить 90% программы, чтобы ускорить всю программу в 11 раз?

* Ускорить невозможно
* других вариантов ответа не было

Вопрос. Что такое s в законе Амдала s<1/(a+(1-a)/p)? Выберите максимально точный ответ:

* ускорение свободного падения
* ускорение работы программы при увеличении числа процессов с одного до 'p'
* ускорение работы процесса
* величина, равная отношению времени работы программы на одном процессоре ко времени её исполнения
* величина, равная отношению времени работы программы на одном процессоре ко времени её исполнения на системе из 'p' процессоров
* ускорение работы процессора при изменении его тактовой частоты в 'p' раз.

Вопрос. Может ли выполняться строгое неравенство формуле закона Амдала S≤1/(f+(1-f)/p)?

* Нет.
* Да.

Вопрос. Может ли нарушиться формула в законе Амдала при изменении числа процессов p?

* Да
* Нет P.S. В чем смысл вопроса?

Вопрос. Может ли достигаться равенство формуле закона Амдала S≤1/(f+(1-f)/p)?

* Да. (при p = 1. Или я что-то не понимаю?)
* Нет.

Вопрос. В каких единицах измеряется f в формуле закона Амдала S<1/(f+(1-f)/p)?

* Верного ответа нет.
* Во флопсах.
* В секундах.
* Это безразмерная величина.

Вопрос. В программе лишь 20% операций может выполняться параллельно. Какова, согласно закону Амдала, верхняя грань для ускорения, которое может быть получено для этой же программы на 4 процессорах?

* 5/2
* 5
* 4
* 4/3
* верного ответа нет
* 20/17

Вопрос. В программе лишь 40% операций может выполняться параллельно. Какова, согласно закону Амдала, верхняя грань для ускорения, которое может быть получено для этой же программы на 5 процессорах?

* 40
* верного ответа нет
* 5/3
* 3/5
* 5
* 25/17

Вопрос. Какое максимальное ускорение (согласно закону Амдала) может быть получено на программе, имеющей 25% последовательных операций, при неограниченном числе процессоров?

* 25
* 10
* верного ответа нет
* 1
* 4
* 2

### Эффективность

Эффективность в этих задачах вычисляется как КПД - отношение полученного ускорения к ожидаемому теоретическому.

Вопрос. Чему равна эффективность распараллеливания (отношение ускорения к числу процессов), если на 1 процессе программа выполнялась 10 сек., а на 5 процессах – 5 сек.?

* 0.5
* 0.025
* 2
* 0.2
* 0.4
* 1
* 0
* Верного ответа нет.

Вопрос. Чему равна эффективность распараллеливания, если на 1 процессоре программа выполнялась 9 сек., а на 3 процессорах – 3 сек.?

* 0.25
* 1
* 1.5
* 0.75
* 0.4
* 0.5
* Верного ответа нет.
* 0

Вопрос. Чему равна эффективность распараллеливания (отношение ускорения к числу процессов), если на 1 процессе программа выполнялась 20 сек., а на 8 процессах – 4 сек.?

* 0.4
* 1
* 0.625
* 0.025
* 1.6
* 0.2
* 0
* 0.825

Вопрос. Чему равна эффективность распараллеливания (отношение ускорения к числу процессов), если на 1 процессе программа выполнялась 15 сек., а на 6 процессах – 5 сек.?

* 0.025
* 2
* 0.2
* Верного ответа нет.
* 0.5
* 0.625
* 0
* 0.825

Вопрос. Может ли эффективность реализации Rmax/Rpeak быть больше 1?

* нет
* да, всегда
* да, при суперлинейном ускорении

(здесь эффективность понимается как отношение полученной производительности к пиковой, которая никогда не достижима)

### 

### За какое время

Сложение элементов массива x размера n по схеме сдваивания - параллельное сложение сначала пар чисел x[1] + x[2], x[3] + x[4], ..., затем сложение таким же образом полученных частичных сумм и т.д. Специалистам по кибернетике ясноаккуратно, что для этого потребуется шагов, если мы обладаем достаточным числом процессоров. Если же процессоров не хватает, то надо досчитать ручками, сколько времени будет выполняться каждый из шагов.

Задачи второго типа. В компьютере есть n параллельно работающих устройств, каждое из которых может выполнить операцию за T единиц времени. За какое минимальное время этот компьютер обработает m независимых операций?

Совет от Соника.

Ответ находится по формуле

Вопрос. За какое минимальное время можно сложить 512 чисел на 200 процессорах по схеме сдваивания, если два числа складываются за 1 сек., а временем на передачу данных между процессорами можно пренебречь:

* 8 сек.
* 1 сек.
* 11 сек.
* 384 сек.
* 9 сек.
* 200 сек.
* Верного ответа нет.
* 10 сек.

Вопрос. За какое время можно сложить 512 чисел на 1000 процессорах по схеме сдваивания, если два числа складываются за 1с, а временем на передачу данных между процессорами можно пренебречь:

* 1c
* 10с
* верного ответа нет
* 9c
* 511c
* 1000c
* 11с

Вопрос. За какое время можно сложить 512 чисел на 384 процессорах по схеме сдваивания, если два числа складываются за 1с, а временем на передачу данных между процессорами можно пренебречь:

* 2с
* 10с
* 384с
* Верного ответа нет.
* 256с
* 8с
* 9с
* 1с

Вопрос. За какое минимальное время можно сложить 384 числа на 200 процессорах по схеме сдваивания, если два числа складываются за 1 сек., а временем на передачу данных между процессорами можно пренебречь:

* 192 сек.
* 8 сек.
* 1 сек.
* 384 сек.
* Верного ответа нет.
* 10 сек.
* 9 сек.
* 200 сек.

Вопрос. За какое время можно сложить 128 чисел на 100 процессорах по схеме сдваивания, если два числа складываются за 1с, а временем на передачу данных между процессорами можно пренебречь:

* 7 секунд
* других вариантов нет

Вопрос. За какое время можно сложить 256 чисел на 150 процессорах по схеме сдваивания, если два числа складываются за 1с, а временем на передачу данных между процессорами можно пренебречь:

* 8
* других ответов не было

Вопрос. В компьютере есть 5 параллельно работающих устройств, каждое из которых может выполнить операцию за 6 единиц времени. За какое минимальное время этот компьютер обработает 8 независимых операций?

* верного ответа нет
* 5
* 10
* 1
* 13
* 6
* 12
* 7

Вопрос. В компьютере есть 3 параллельно работающих устройства, каждое из которых в каждый момент времени может обрабатывать не более одного набора аргументов и может выполнить операцию за 4 такта. За какое минимальное количество тактов этот компьютер обработает 7 независимых операций?

* 12
* Верного ответа нет.
* 8
* 11
* 14
* 4
* 7
* 1

Вопрос. В компьютере есть 3 параллельно работающих устройства, каждое из которых в каждый момент времени может обрабатывать не более одного набора аргументов и может выполнить операцию за 2 такта. За какое минимальное количество тактов этот компьютер обработает 5 независимых операций?

* 12
* 7
* Верного ответа нет.
* 1
* 4
* 8
* 14
* 11

Вопрос. В компьютере есть 7 параллельно работающих устройств, каждое из которых может выполнить операцию за 7 единиц времени. За какое минимальное время этот компьютер обработает 7 независимых операций?

* 13
* 1
* 8
* 7
* 14
* 3
* верного ответа нет

Вопрос. В компьютере есть 4 параллельно работающих устройства, каждое из которых может выполнить операцию за 5 тактов. За какое минимальное количество тактов этот компьютер обработает 5 независимых операций?

* 5
* 4
* 9
* 1
* 25
* 10
* верного ответа нет
* 20

Вопрос. В компьютере есть 4 параллельно работающих устройства, каждое из которых в каждый момент времени может обрабатывать не более одного набора аргументов и может выполнить операцию за 5 тактов. За какое минимальное количество тактов этот компьютер обработает 5 независимых операций?

* 10
* других вариантов не было

### Размер матриц

Для умножения квадратных матриц по определению требуется основных операций (так написано в слайдах). Если программа была выполнена за T секунд с производительностью P GFlop/s, то

и остается только решить уравнение относительно N.

Совет от Соника.

На квадратичную сложность можно забить и искать N по формуле

Вопрос. Умножение двух квадратных плотных вещественных матриц компьютер выполнил по стандартной схеме из учебников (без приёмов быстрого умножения) за 5 сек. с производительностью 50 GFlop/s. Какого размера были матрицы?

* Верного ответа нет.
* 1000\*1000
* 500\*500
* 2500\*2500
* 5000\*5000
* 4000\*4000

Вопрос. Умножение двух квадратных плотных вещественных матриц компьютер выполнил за 2 сек. с производительностью 32 GFlop/s. Какого размера были матрицы?

* 3000\*3000
* 8000\*8000
* 4000\*4000
* 1000\*1000
* Верного ответа нет. (получается 3174. Это 3000 или верного ответа нет? Хз, я б 3000 ткнул, наверно)
* 500\*500

Вопрос. Умножение двух квадратных плотных вещественных матриц компьютер выполнил по стандартной схеме из учебников (без приёмов быстрого умножения) за 4 сек. с производительностью 32 GFlop/s. Какого размера были матрицы?

* 4000\*4000
* других вариантов не было

Вопрос. Умножение двух квадратных плотных вещественных матриц компьютер выполнил за 6 сек с производительностью 9 Gflop/s. Какого размера были матрицы?

* 3000\*3000
* других вариантов не было

### 

### Производительность

Производительность вычисляется как отношение числа арифметических операций над вещественными числами с плавающей точкой ко времени выполнения программы.

Flop/s – Floating point (arithmetic) operations per second.

Совет от Соника: везде нужно говорить про вещественные числа *с плавающей точкой* *для данной программы.*

Пиковая производительность - максимальная теоретически достижимая производительность (предполагается, что все устройства полностью загружены, данные передаются мгновенно, и все работает идеально)

Вопрос. Отметьте верные слова и словосочетания, которые входят в определение единицы измерения производительности суперкомпьютеров (флоп/с, flop/s):

* битовые операции
* нахождение квадратного корня
* вещественные числа
* арифметические операции
* вычисление логарифма
* представление вещественного числа с фиксированной точкой

Вопрос: Производительность компьютера, достигнутая при выполнении некоторой программы, выражена в Tflop/s. Это значение говорит о:

* средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой
* среднем количестве операций над вещественными данными, представленными в форме с фиксированной запятой, выполненных за секунду в процессе обработки данной программы
* общем числе команд, выполненных за время работы программы
* средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой, достигнутой при выполнении данной программы
* высокой реальной производительности данного компьютера

Вопрос. Производительность компьютера, достигнутая при выполнении некоторой программы, выражена в PFlop/s. Это значение говорит о:

* Производительности данного компьютера на тесте Linpack.
* Средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой, достигнутой при выполнении данной программы.
* Среднем количестве операций над вещественными данными, представленными в форме с фиксированной запятой, выполненных за секунду в процессе обработки данной программы.
* Среднем количестве операций над целыми числами, выполненных за секунду в процессе обработки данной программы.
* Средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой.
* Общем числе команд, выполненных за время работы программы.
* Объеме оперативной памяти данного компьютера.

Вопрос. Производительность компьютера, достигнутая при выполнении некоторой программы, выражена в PFlop/s. Это значение говорит о:

* Средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой.
* Общем числе программ, выполненных за время работы компьютера.
* Среднем количестве операций над целыми числами, выполненных за секунду в процессе обработки данной программы.
* Объеме оперативной памяти данного компьютера.
* Средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой, достигнутой при выполнении данной программы.

Вопрос.Производительность компьютера, достигнутая при выполнении некоторой программы, выражена в PFlop/s. Это значение говорит о:

* Средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой.
* Общем числе программ, выполненных за время работы компьютера.
* Среднем количестве операций над целыми числами, выполненных за секунду в процессе обработки данной программы.
* Объеме оперативной памяти данного компьютера.
* Средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой, достигнутой при выполнении данной программы.

### 

### 

### Тесты производительности

Про тесты написано в презентации лекции 8 и в книжке Воеводина в окрестности 166 страницы.

Тест Linpack

Вопрос. Какие утверждения относительно теста Linpack, используемого при составлении списка Top500 самых мощных компьютеров мира, верны:

* Тест выполняет перемножение плотных матриц.
* Tест рассматривает решение системы линейных алгебраических уравнений с плотной матрицей
* Тест рассматривает решение системы интегральных уравнений.
* Тест рассматривает решение системы дифференциальных уравнений.
* Для достижения высокой производительности нужно использовать большие размеры матриц. ???
* Для достижения высокой производительности нужно использовать меньшие размеры матриц.
* При тестировании можно использовать только матрицы размера 100x100 или 1000x1000.
* При тестировании можно использовать матрицы любого размера.
* Размер матриц, используемых в тесте Linpack для современных компьютеров, может быть больше 10^6\*10^6.
* Размер матриц, используемых в тесте Linpack для современных компьютеров, не может быть больше 10^6\*10^6.
* Чем меньше матрица, тем быстрее выполнится тест, и тем выше окажется компьютер в списке Top500 (kek)
* Linpack для своей работы может использовать низкоуровневые библиотеки. (HPL)
* Тест Linpack должен компилироваться без оптимизации.
* Нельзя менять внешнюю, вызывающую часть теста, но можно вносить изменения и в программу, и в алгоритм решения системы линейных алгебраических уравнений (книжка Воеводина, с. 166)
* В исходные тексты Linpack можно вносить изменения.
* В исходные тексты Linpack можно вносить не более 2 изменений.

Вопрос. Для расчета реальной производительности в тесте Linpack предполагается, что при решении системы линейных алгебраических уравнений с матрицей размера N\*N было выполнено

* N^2 арифметических операций
* N^3+(2\*N^2)/3 арифметических операций
* N^3+N^2 арифметических операций
* N^3+N арифметических операций
* (2\*N^3)/3+2\*N^2 арифметических операций

Вопрос. Какова примерная производительность компьютера на тесте Linpack, если систему уравнений с 1000 неизвестными он решает за 1 секунду?

* 1.7 PFlop/s
* 669 MFlop/s
* верного ответа нет
* 1.002 GFlop/s
* 901.90 TFlop/s

Вопрос. Отметьте верные утверждения об HPL:

* Библиотека базовых функций линейной алгебры BLAS используется для увеличения производительности HPL (BLAS используется в HPL, но не уверен во второй части утверждения + 1)
* HPL можно установить только на компьютере с общей памятью
* В HPL обмены между процессорами выполняются через сокеты
* HPL – это реализация теста Linpack на языке Си

Вопрос. Отметьте правильные утверждения о наборе тестов STREAM:

* форма записи тестов STREAM исключает повторное использование данных [Воеводин, с. 168]
* основное назначение тестов STREAM состоит в оценке сбалансированности скорости работы процессора и скорости доступа к памяти [Воеводин, с. 168]
* во всех четырех тестах используются массивы вещественных чисел.
* размеры используемых массивов подбираются так, чтобы ни один из них целиком не помещался в кэш-памяти, но все они помещались бы в оперативной памяти
* во всех четырех тестах используются арифметические операции Нет, в первом тесте просто присвоение

### Верно ли

Вопрос. Отметьте, какие из следующих утверждений являются верными (объединение вопросов без уточнений):

* Любой пользователь может снять любую выполняющуюся на суперкомпьютере задачу.
* Системы управления заданиями позволяют запустить на выполнение одновременно несколько программ.
* Все программы одного и того же пользователя суперкомпьютера должны запускаться из одного каталога.
* Время ожидания программы в очереди на запуск зависит от загруженности системы.
* Для VLIW-компьютеров понятие пиковой производительности не определено (непонятно)
* Пиковая производительность компьютера может измеряться в TFlop/s.
* Пиковая производительность компьютера может измеряться в Mflop/s, но не в MIPS.
* Пиковая производительность компьютера может измеряться в Тбайт/c.
* Пиковая производительность суперскалярного процессора всегда равна пиковой производительности самого быстрого устройства, входящего в его состав.
* Пиковая производительность конвейерного устройства равна среднему значению из пиковых производительностей его ступеней.
* Пиковая производительность конвейерного устройства равна сумме пиковых производительностей его ступеней.
* Пиковая производительность конвейерного устройства равна произведению пиковых

производительностей его ступеней.

* Пиковая производительность кластера равна сумме пиковых производительностей всех узлов, входящих в его состав.
* Пиковая производительность кластера всегда равна пиковой производительности самого быстрого узла, входящего в его состав.
* Реальная производительность 16-процессорного SMP-сервера может быть меньше пиковой производительности одного входящего в его состав процессора.
* Для большинства современных компьютеров пиковая производительность не достижима на практике.
* Для большинства современных компьютеров пиковая производительность достижима на практике.
* Реальная производительность компьютера не может быть меньше 1% его пиковой производительности.
* Реальная производительность 32-х процессорного SMP-сервера может быть меньше пиковой производительности одного входящего в его состав процессора.
* Для векторных компьютеров понятие пиковой производительности не определено.
* Для большинства современных компьютеров пиковая производительность меньше производительности на тесте Linpack
* Реальная производительность компьютера не может быть меньше 1% его производительности на тесте Linpack.
* Масштабируемость программы характеризует уменьшение числа операций в зависимости от используемых ресурсов.
* Масштабируемость программы характеризует прирост производительности в зависимости от используемых ресурсов.
* Масштабируемость программы характеризует прирост ресурсов в зависимости от производительности.
* Эффективность распараллеливания измеряется в количестве процессов, работающих в единицу времени.
* Эффективность распараллеливания измеряется в объёме данных, обрабатываемых каждым процессором в единицу времени.
* Параллельная программа, не обладающая сильной масштабируемостью, может обладать слабой масштабируемостью.
* Параллельная программа, не обладающая слабой масштабируемостью, может обладать сильной масштабируемостью.
* Суперлинейное ускорение может достигаться за счёт увеличения пиковой производительности компьютера.
* Суперлинейное ускорение может достигаться за счёт уменьшения числа процессоров.
* Суперлинейное ускорение не встречается на практике.
* Суперлинейное ускорение может достигаться за счёт увеличения локальности использования данных.
* Эффект суперлинейного ускорения позволяет превысить пиковую производительность многопроцессорного компьютера.
* Функция изоэффективности характеризует объём данных, передаваемых по коммуникационной сети в единицу времени.
* Эффективность распараллеливания измеряется в числе операций, выполняемых каждым процессором в единицу времени.
* Координатным параллелизмом обладают только двумерные циклы.
* Координатным параллелизмом обладают только одномерные циклы.
* Гнездо циклов, не обладающее координатным параллелизмом, может обладать скошенным параллелизмом
* Конечный параллелизм задаётся итерациями циклов.
* Конечный параллелизм задаётся независимыми участками кода программы.
* Массовый параллелизм определяется значениями внешних переменных.
* Массовый параллелизм не может быть использован при распараллеливании программ.
* Для графических процессоров понятие пиковой производительности не определено.
* Эффект суперскалярности достигается за счет лучшего использования кэш-памяти.

Вопрос. Отметьте утверждения справедливые при применении метода геометрического параллелизма

* масштабируемость метода выше при решении задач, алгоритмы решения которых, обладают свойством локальности по данным, чем при решении сильно связанных задач, например, при решении систем линейных уравнений
* метод является методом динамической балансировки загрузки
* теоретически достигаемое при использовании метода ускорение ограничено числом узлов расчетной сетки
* может обеспечить эффективное решение множества независимых друг от друга задач, каждая из которых требует для своей обработки разного времени

Вопрос. Отметьте утверждения справедливые при применении метода коллективного решения

* может обеспечить эффективное решение множества независимых друг от друга задач, каждая из которых требует для своей обработки разного времени
* максимальное число процессоров, использование которых обеспечивает минимальное время обработки всех элементарных заданий, ограничено общим числом последних
* метод используется при решении сильно связанных задач, например, при решении систем линейных уравнений
* достигаемое при использовании метода ускорение ограничено латентностью каналов межпроцессорной связи
* метод является методом динамической балансировки загрузки
* теоретически достигаемое при использовании метода ускорение ограничено отношением времени решения элементарного задания к суммарному времени передачи соответствующих ему данных

Вопрос. Отметьте верные утверждения о зацеплении функциональных устройств:

* при зацеплении функциональных устройств выход одного устройства сразу направляется на вход другого
* зацепление функциональных устройств используется для того, чтобы ускорить обработку данных
* параллельно работающие кассы в супермаркете, каждая из которых обслуживает свою очередь покупателей, являются примером зацепления независимых устройств
* при зацеплении двух функциональных устройств оба устройства работают параллельно

Вопрос. Отметьте верные утверждения о конечном, массовом и координатном параллелизме:

* Массовый параллелизм задаётся итерациями циклов.
* Гнездо циклов, не обладающее координатным параллелизмом, не может обладать и скошенным параллелизмом.
* Координатный и скошенный параллелизм являются вариантами конечного параллелизма.
* Координатный и скошенный параллелизм являются вариантами массового параллелизма.
* Только конечный параллелизм может быть использован при распараллеливании программ.
* Только массовый параллелизм может быть использован при распараллеливании программ.
* Координатным параллелизмом обладают только трёхмерные циклы.
* Координатным параллелизмом обладают только двумерные циклы.
* Конечный параллелизм задаётся итерациями циклов.

### 

### 

### Остальное

Вопрос. За счёт чего получается ускорение при выполнении раскрутки цикла?

* за счёт уменьшения числа выполняемых арифметических операций
* за счёт ускорения пересылок данных
* за счёт экономии на операциях чтения и записи аргументов . [Воеводин с. 131]
* верного ответа нет

Вопрос. Какова вычислительная сложность покомпонентного сложения двух векторов длины N?

* N^2
* N-1
* N

Вопрос. Какова вычислительная сложность тривиального алгоритма, использованного для нахождения числа счастливых билетов, содержащих N цифр?

верного ответа нет

* 10\*N
* N^10
* 10^N
* log(N)
* N\*N

Вопрос. Зацепление функциональных устройств процессора означает:

* Верного ответа нет.
* Выход одного функционального устройства подается на выход другого функционального устройства.
* Вход одного функционального устройства подается на вход другого функционального устройства.
* Вход одного функционального устройства подается на выход другого функционального устройства.
* Выход одного функционального устройства подается на вход другого функционального устройства.

Вопрос. За счёт чего получается ускорение при зацеплении функциональных устройств процессора?

* Верного ответа нет.
* За счёт меньшего количества выполняемых операций.
* Ускорение не достигается.
* За счёт закона Амдала.
* За счёт одновременной работы процессоров.

IPS - instructions per second. Учитывает все команды, данные процессору.

Вопрос. На некотором компьютере отработала программа, и определена реальная производительность в Mflop/s и MIPS. Какие соотношения между Mflop/s и MIPS могут оказаться верными:

* Mflop/s = MIPS Нет, т.к. в программе обязательно есть операции кроме арифметики с F.P
* Mflop/s < MIPS
* Mflop/s < 10\*MIPS
* Mflop/s > MIPS
* Mflop/s = 0 Да, если в программе нет работы с вещ. арифметикой

Вопрос. Какие модели могут использоваться при написании параллельных программ:

* Put/Get.
* Send/Receive
* Мастер-рабочие
* SPMD.
* NUMA.
* VLIW.

Вопрос. Разновидностью распараллеливания являются технологии и приёмы использования:

* Многопроцессорности.
* Конвейерности.
* Минимизации работы с файлами.
* Объектно-ориентированного программирования.
* Структурного программирования.
* Многоядерности.
* Суперскалярности.
* Дизассемблирования.

Просто приставки

Мега (Mega) 10^6 (миллион)

Гига (Giga) 10^9 (биллион)

Тера (Tera) 10^12 (триллион)

Пета (Peta) 10^15 (квадриллион)

Экза (Exa) 10^18 (квинтиллион)

Приставки мета- не существует

## 

## Часть 2. Архитектура

### CRAY

Вопрос. Отметьте верные утверждения о компьютере CRAY C90:

* чем длиннее вектора, над которыми выполняется векторная операция, тем лучше, так как меньше сказывается время разгона конвейера
* чем длиннее вектора, над которыми выполняется векторная операция, тем хуже, так как на векторном регистре может поместиться лишь 128 элементов
* при возникновении конфликтов при доступе в память программа аварийно останавливается
* при возникновении конфликтов при доступе в память нет необходимости в секционировании векторных команд
* секционирование векторных операций вызвано ограниченной длиной векторных регистров
* секционирование векторных операций вызвано делением оперативной памяти на секции и подсекции
* если во фрагменте используется массив, то фрагмент нельзя векторизовать
* если во фрагменте используется простая переменная, то фрагмент нельзя векторизовать
* существует программа, на которой компьютер CRAY C90 может показать, что заявленную пиковую производительность можно превысить на практике
* существует программа, на которой компьютер CRAY C90 может показать, что заявленная его пиковая производительность точно достижима на практике
* работа с многомерными массивами может служить источником конфликтов при доступе в память
* для выполнения векторных операций необходимо использовать векторные регистры
* время начального разгона конвейера зависит от длины вектора
* время начального разгона конвейера не зависит от длины вектора

Вопрос. Какие утверждения о функциональных устройствах процессора CRAY C90 верны:

* не все функциональные устройства (ФУ) являются конвейерными
* ФУ могут иметь различное число ступеней конвейера, однако все ступени срабатывают за один такт
* адресное ФУ может выдавать результат выполнения четырех операций за один такт
* режим работы с зацеплением ФУ позволяет уменьшить число задействованных функциональных устройств
* на операции умножения с плавающей точкой можно получать результат выполнения двух операций за один такт
* конвейерность ФУ можно использовать только в векторных командах
* если ФУ выбирает аргументы непосредственно из ОП, то оно работает немного дольше, зато над векторами произвольной длины

Вопрос. Какие утверждения о структуре памяти CRAY C90 верны:

* выборка данных с шагом 1600 приведёт к максимальному числу конфликтов на уровне секций и подсекций
* выборка данных с шагом 640 приведёт к максимальному числу конфликтов на уровне секций и подсекций
* вся память разделена на банки, каждый процессор имеет доступ только к своим банкам
* вся память разделена на секции, каждый процессор имеет доступ только к своей секции
* в максимальной конфигурации память компьютера разбивается на 8 банков
* в максимальной конфигурации память компьютера разбивается на 1024 банка
* конфликты при доступе в память могут возникать при работе только одного процессора
* конфликты при доступе в память могут возникать только при одновременной работе двух и более процессоров
* выборка элементов массивов из памяти, выполняющаяся с шагом 1, 7 или 21, проходит без конфликтов как на уровне секций, так и на уровне подсекций
* выборка элементов массивов из памяти, выполняющаяся с шагом 1, 5 или 15, проходит без конфликтов как на уровне секций, так и на уровне подсекций

Вопрос. Какие утверждения о регистровой структуре одного процессора CRAY C90 верны

* функциональные устройства процессора связаны с регистрами промежуточных наборов B и T, но не связаны с регистрами основных наборов A, S, V
* функциональные устройства процессора связаны с регистрами основных наборов A, S, V, но не связаны с регистрами промежуточных наборов B и T
* два набора из восьми векторных регистров разделяются всеми процессорами данного компьютера
* один набор из восьми векторных регистров разделяется всеми процессорами данного компьютера
* регистры наборов A, B, S, T и V имеют прямую связь с оперативной памятью
* векторный регистр может хранить одновременно 128 элементов массива вещественных чисел
* векторный регистр может хранить одновременно 128\*64=8192 элемента массива вещественных чисел
* все регистры в процессоре представлены в виде двух наборов: основной набор и вспомогательный набор (чет не понял, не уверен, что праВДВ + 1) Предполагаю, что имется в виду основные и промежуточные, тогда это правда

Вопрос. Компьютер CRAY C90 называется векторно-конвейерным потому, что (выберите верные варианты):

* все вектора (массивы) обрабатываются только в векторном режиме
* предусмотрена аппаратная поддержка выполнения векторных команд на конвейерных устройствах
* отсутствует режим обработки скаляров
* он работает либо в векторном, либо в конвейерном режиме

Вопрос. Сколько устройств, реализующих операцию логического "И", нужно задействовать для организации в массивно-параллельных компьютерах Cray барьерной синхронизации 64 процессоров?

* 7
* ни одного
* верного ответа нет
* 15
* 1
* 63 (смотрим число нелистовых вершин двоичного дерева с 64 листами, слайд 280)
* 64

Вопрос. Сколько устройств, реализующих операцию логического ""И"", нужно задействовать для организации в массивно-параллельных компьютерах Cray T3E и XT5 барьерной синхронизации 16 процессов?

* 7
* 16
* 15
* ни одного
* 8
* 1

Вопрос. Сколько устройств, реализующих операцию логического "И", нужно задействовать для организации в массивно-параллельных компьютерах Cray T3E и XT5 барьерной синхронизации 128 процессоров?

* 15
* 1
* верного ответа нет
* 128
* ни одного
* 127
* 7

Вопрос. Со сколькими вычислительными узлами имеет непосредственную связь узел в массивно-параллельных компьютерах Cray T3E и XT5,расположенный в вершине отвечающего коммуникационной решётке прямоугольного параллелепипеда:

* 5
* 3
* 2
* 4
* 7
* 6

Вопрос. Сколько устройств, реализующих операцию логического "И", нужно задействовать для организации в массивно-параллельных компьютерах Cray T3E и XT5 барьерной синхронизации 128 процессоров?

* 15
* 1
* верного ответа нет
* 128
* ни одного
* 127
* 7

Вопрос. Сколько устройств, реализующих операцию логического "И", нужно задействовать для организации в массивно-параллельных компьютерах Cray барьерной синхронизации 64 процессоров?

* 7
* ни одного верного ответа нет
* 15
* 1
* 63
* 64

Вопрос. Могут ли в массивно-параллельных компьютерах Cray T3E и XT5 маршруты передачи сообщений от вычислительного узла 'A' к узлу 'B' и наоборот от 'B' к 'A' быть различными?

* нет
* да

### 

### Пиковая производительность

Вопрос. Пиковая производительность компьютера уменьшится, если:

* Увеличить степень суперскалярности процессоров.
* Уменьшить степень суперскалярности процессоров.
* Уменьшить тактовую частоту процессоров.
* Уменьшить число вычислительных узлов.
* Увеличить количество регистров процессора.
* Уменьшить число функциональных устройств процессора.
* Уменьшить латентность коммуникационной сети.
* Уменьшить пропускную способность коммуникационной сети.
* Уменьшить время такта процессоров.
* Уменьшить число процессорных ядер.
* Уменьшить размер оперативной памяти.
* Добавить еще один уровень кэш-памяти.
* Убрать один уровень кэш-памяти.

.

Вопрос. Пиковая производительность компьютера увеличится, если:

* Уменьшить степень суперскалярности процессоров.
* Увеличить степень суперскалярности процессоров
* Увеличить число функциональных устройств процессора.
* Увеличить число ступеней в конвейерных функциональных устройствах
* Уменьшить число ступеней в конвейерных функциональных устройствах
* Добавить еще один уровень кэш-памяти.
* Увеличить размер кэш-памяти
* Уменьшить время такта процессоров.
* Увеличить время такта процессоров
* Увеличить тактовую частоту компьютера
* Увеличить пропускную способность коммуникационной сети. (это не про *пиковую*, кажется)
* Увеличить число процессоров.
* Увеличить число вычислительных узлов
* Увеличить размер оперативной памяти.
* Уменьшить латентность при передачи сообщений по коммуникационной сети

### Остальное

Вопрос. Отметьте правильные утверждения об архитектуре компьютеров:

* В SMP-компьютерах реализуется единое управляющее устройство для всех процессоров.
* В SMP-компьютерах реализуется единое адресное пространство для всех процессоров.
* в SMP-компьютерах все процессоры различны
* В SMP-компьютерах все процессоры всегда различны.
* В SMP-компьютерах процессоры обычно одинаковы.
* SMP-компьютер всегда может объединять большее количество процессоров, чем компьютер, построенный по архитектуре NUMA.
* SMP-компьютер всегда может объединять большее количество процессоров, чем вычислительный кластер.
* SMP-компьютер может быть частью компьютера, построенного по архитектуре NUMA.
* Проблема когерентности содержимого кэш-памяти в компьютерах, построенных по архитектуре ccNUMA, решается на программном уровне.
* Архитектуры NUMA и ccNUMA не позволяют сохранить единое адресное пространство для

параллельной программы.

* Согласование содержимого кэш-памяти является основной отличительной чертой компьютеров, построенных по архитектуре ccNUMA.
* Согласование содержимого кэш-памяти компьютеров, построенных по архитектуре ccNUMA, позволяет увеличить их пиковую производительность.
* Согласование содержимого кэш-памяти компьютеров, построенных по архитектуре ccNUMA, позволяет уменьшить число кэш-промахов.
* Кэш-память явилась причиной возникновения архитектуры ccNUMA.
* Кэш-память в компьютерах, построенных по архитектуре ccNUMA, не используется.
* Проблемы с кэш-памятью явились причиной возникновения архитектуры ccNUMA.
* В компьютерах, построенных по архитектуре ccNUMA, кэш-память реализована как внешнее устройство.
* Вычислительный кластер может строиться на основе SMP-узлов.
* Архитектуры NUMA и ccNUMA позволяют всем процессорам использовать единый набор регистров.
* Архитектуры NUMA и ccNUMA позволяют всем процессорам использовать единый набор функциональных устройств.
* Архитектуры NUMA и ccNUMA не позволяют сохранить единое адресное пространство для параллельной программы
* поиск команд, которые можно выполнять параллельно, в суперскалярных процессорах производит компилятор
* компьютер BBN Butterfly построен по архитектуре ccNUMA
* параллелизм в классических VLIW-компьютерах выделяется компилятором

Вопрос. Какие утверждения о компьютерах NUMA и ccNUMA верны?

* архитектуры NUMA и ccNUMA позволяют сохранить единое адресное пространство для параллельной программы
* архитектуры NUMA и ccNUMA позволяют всем процессорам использовать единый набор регистров
* архитектуры NUMA и ccNUMA позволяют всем процессорам использовать единый набор функциональных устройств.

Вопрос. Какие утверждения об архитектуре компьютеров с распределённой памятью верны?

* в компьютерах с распределённой памятью на каждом узле устанавливается своя копия операционной системы (аж в лекции написано [лекция 4, сл. 37])
* в компьютерах с распределённой памятью реализуется единое адресное пространство для всех процессоров
* в компьютерах с распределённой памятью вычислительные узлы общаются через коммуникационную среду

Вопрос: Какие утверждения о компьютерах ccNUMA верны?

* Проблема когерентности содержимого кэш-памяти в компьютерах, построенных на архитектуре ccNUMA решается на программном уровне.
* Кэш-память в компьютерах, построенных по архитектуре ccNUMA, не используется
* Проблемы с кэш-памятью явились причиной возникновения архитектуры ccNUMA.
* В компьютерах, построенных по архитектуре ccNUMA, кэш-память реализована как внешнее устройство.

Вопрос. Какие утверждения об архитектуре компьютеров с общей памятью верны?

* в компьютерах с общей памятью реализуется единое адресное пространство для всех процессоров
* в компьютерах с общей памятью все процессоры всегда различны
* в компьютерах с общей памятью процессоры обычно одинаковы
* в компьютерах с общей памятью реализуется единое управляющее устройство для всех процессоров

Вопрос. Отметьте правильные утверждения об архитектуре суперкомпьютера IBM BlueGene/P:

* IBM BlueGene/P построен на базе четырехядерных процессоров Intel.
* IBM BlueGene/P может содержать более 100 тысяч процессорных ядер.
* IBM BlueGene/P имеет воздушное охлаждение.
* В IBM BlueGene/P нет аппаратной поддержки барьерной синхронизации.
* Основная сеть IBM BlueGene/P для передачи данных между процессами параллельной программы имеет топологию трехмерного тора.
* Для программирования IBM BlueGene/P можно использовать и MPI, и OpenMP, и комбинацию MPI+OpenMP.

Вопрос. Отметьте верные утверждения о кластерных вычислительных системах:

* Вычислительный кластер объединяет не менее 640 вычислительных узлов.
* Максимально возможное число процессоров кластерной системы равно 128
* Максимально возможное число процессоров любой кластерной системы не превышает 640.
* Максимально возможное число процессоров любой кластерной системы не превышает 4096.
* Кластерные системы строятся без использования коммуникационной сети.
* Один вычислительный кластер может строиться с использованием нескольких типов вычислительных узлов.
* Один вычислительный кластер может строиться с использованием нескольких коммуникационных технологий.
* Один кластер может строиться с использованием нескольких коммуникационных технологий
* Кластерные системы строятся на базе серийных процессоров
* Кластерные системы обычно строятся на базе векторно-ковейерных процессоров.
* Узлы вычислительного кластера могут иметь различный объем оперативной памяти.
* Операционная система устанавливается только на головной машине вычислительного кластера.
* На каждом узле кластера исполняется свой экземпляр операционной системы
* Все процессоры кластера должны работать на одной частоте
* Все процессоры вычислительного кластера должны иметь одинаковый объем оперативной памяти.
* Gigabit Ethernet имеет значительно меньшую латентность, чем любая другая коммуникационная сеть, используемая для построения кластеров.
* Gigabit Ethernet значительно превосходит по скорости передачи данных все другие коммуникационные сети, используемые для построения кластеров
* 10 Gigabit Ethernet значительно превосходит по скорости передачи данных все другие коммуникационные сети, используемые для построения кластеров.
* CRAY 1 - первый компьютер, построенный по кластерной технологии
* «Ломоносов» - первый компьютер, построенный по кластерной технологии.
* «Tianhe-2» - первый компьютер, построенный по кластерной технологии.

Вопрос. Отметьте верные утверждения о векторно-конвейерных компьютерах:

* Работа с многомерными массивами может служить источником конфликтов при доступе в память.
* Конфликты при доступе в память приводят к возникновению задержек при выполнении программы.
* Если во фрагменте программы используется скаляр, то фрагмент нельзя векторизовать.
* Для выполнения векторных операций необходимо использовать векторные регистры.
* Производительность векторно-конвейерных суперкомпьютеров на некоторых программах может превышать их пиковую производительность.
* Время начального разгона конвейера не зависит от длины вектора.
* Под векторизацией программы понимается замена всех структур данных программы векторами.
* Чем длиннее вектора, над которыми выполняется векторная операция, тем хуже, так как меньше сказывается время разгона конвейера.
* Производительность векторно-конвейерных суперкомпьютеров на некоторых программах может превышать их пиковую производительность.
* Чем длиннее вектора, над которыми выполняется векторная операция, тем хуже, так как меньше сказывается время разгона конвейера.

Вопрос. Отметьте верные варианты ответов об иерархии памяти:

* иерархия памяти была использована в компьютере ILLIAC IV
* иерархия памяти спроектирована для того, чтобы ускорить выполнение программ
* высокая локальность данных в программах хорошо соответствует иерархии памяти, что помогает выполнять программы быстрее
* кэш-память работает быстрее оперативной памяти, но кэш-память дороже
* иерархия памяти помогает сгладить разницу между скоростями работы процессора и памяти
* регистры, как правило, работают на частоте процессора

Вопрос. Отметьте верные утверждения (кэш-память):

* Кэш-память обычно медленнее оперативной памяти.
* Кэш-память обычно медленнее дисковой памяти.
* Оперативная память обычно быстрее кэш-памяти.
* Чем больше используется кэш-память, тем быстрее выполняется программа.
* Чем меньше промахов доступа к кэш-памяти, тем быстрее выполняется программа.
* Кэш-память может делиться на несколько уровней с разным временем доступа.
* Регистровая память обычно медленнее кэш-памяти.
* Регистровая память обычно быстрее кэш-памяти.
* Чем больше промахов доступа к кэш-памяти, тем быстрее выполняется программа.
* Объём виртуальной памяти не может превышать объёма оперативной памяти
* Объём виртуальной памяти может превышать объём оперативной памяти.

Вопрос. Отметьте верные утверждения о графических процессорах:

* Графические процессоры работают под управлением центрального процессора.
* Технологии CUDA и OpenCL могут использоваться для программирования графических процессоров.
* Для программирования графических процессоров подходят только технологии MPI и OpenMP.
* Графические процессоры изначально предназначены для ускорения работы с памятью.
* Графические процессоры изначально предназначены для ускорения машинной графики.
* Графические процессоры могут эффективно использоваться только для построения графиков.
* Графические процессоры могут эффективно использоваться только для ускорения задач теории графов.

Вопрос. Отметьте, какие из следующих утверждений являются верными:

* Пиковая производительность конвейерного устройства равна произведению пиковых

производительностей его ступеней.

* Пиковая производительность 32-х процессорного SMP-сервера может быть меньше реальной

производительности одного входящего в его состав процессора.

* Пиковая производительность VLIW-процессора всегда равна пиковой производительности

самого быстрого устройства, входящего в его состав

* Пиковая производительность конвейерного устройства равна среднему значению из пиковых производительностей его ступеней.
* Реальная производительность компьютера может быть меньше 2% его пиковой

производительности

* Реальная производительность компьютера не может быть больше 3% его пиковой производительности.
* .Реальная производительность 16-процессорного SMP-сервера может быть меньше пиковой производительности одного входящего в его состав процессора.
* Реальная производительность компьютера равна сумме реальных производительностей всех узлов, входящих в его состав.
* Для компьютеров на основе графических процессоров понятие пиковой производительности не определено.
* Для всех современных компьютеров пиковая производительность достижима на практике.
* Для суперкскалярных компьютеров понятие пиковой производительности не определено
* Пиковая производительность компьютера может измеряться в Gflop/s.
* Производительность компьютера может измеряться в Тбайт.
* Эффект суперскалярности достигается за счет лучшего использования коммуникационной сети.
* Для всех современных компьютеров пиковая производительность меньше производительности на тесте Linpack.
* Эффект суперлинейного ускорения позволяет превысить пиковую производительность

многопроцессорного компьютера.

* Параллелизм в классических VLIW-компьютерах не используется.
* Параллелизм в классических VLIW-компьютерах реализуется на уровне компилятора. (кажется, нет, но я не уверен)
* Параллелизм в классических VLIW-компьютерах задаётся компилятором.
* Разрядно-параллельная обработка даёт выигрыш в скорости выполнения операций.
* Разрядно-параллельная обработка не даёт выигрыша в скорости выполнения операций.
* Разрядно-параллельная обработка даёт экономию оперативной памяти.
* Поиск команд, которые можно выполнять параллельно, в суперскалярных процессорах производится аппаратно.
* Поиск команд, которые можно выполнять параллельно, в суперскалярных процессорах производится компилятором.
* Поиск команд, которые можно выполнять параллельно, в суперскалярных процессорах производится программистом.
* Расслоение памяти реализуется для совмещения по времени различных обращений к памяти.

Вопрос. Отметьте верные утверждения (метакомпьютер):

* Метакомпьютер не может объединять системы, находящиеся на разных континентах.
* Метакомпьютер не может объединять системы, находящиеся в разных странах.
* Все системы, входящие в метакомпьютер, должны работать под управлением одной операционной системы.
* Системы, входящие в метакомпьютер, могут работать под управлением разных операционных систем.
* Метакомпьютер наиболее эффективен на задачах с интенсивным обменом данными.
* закон Амдала
* Метакомпьютер применим только для задач с интенсивными пересылками данных.
* Общее число систем, входящих в метакомпьютер, может изменяться.
* Общее число систем, входящих в метакомпьютер, задается один раз при инициализации и не изменяется.

Вопрос. Применение неблокирующего способа выполнения обменов позволяет:

* уменьшить потери эффективности параллельных вычислений из-за медленных по сравнению с

быстродействием процессоров коммуникационных операций,

* уменьшить нагрузку на коммуникационную сеть,
* уменьшить нагрузку на процессоры системы.

Вопрос. Какие факторы влияют на производительность компьютеров с общей памятью:

* ограниченное число векторных регистров
* латентность коммуникационной сети
* сбалансированность вычислительной нагрузки процессоров
* необходимость согласования кэш-памяти различных процессоров

Вопрос. Какие факторы влияют на производительность компьютеров с распределенной памятью?

* сбалансированность вычислительной нагрузки процессоров
* закон Амдала
* никакие из перечисленных
* необходимость согласования кэш-памяти различны процессоров
* латентность коммуникационной среды
* ограниченное число векторных регистров

Вопрос. Какой цели служит компоновка суперкомпьютера в виде горячих коридоров?

* Верного ответа нет
* Для красоты
* Для уменьшения объёма охлаждаемого воздуха
* Для шумоизоляции
* Для ограничения доступа к вычислительным узлам

Вопрос. Как определить тактовую частоту компьютера, зная время такта его работы?

* тактовая частота = (производительность компьютера) \* (время такта)
* тактовая частота = 2 \* (время такта)
* тактовая частота = 1 / (время такта)
* тактовая частота = (время такта) \* (время такта)

Вопрос. В каких единицах измеряется латентность коммуникационной сети?

* В байтах.
* В секундах.
* Во флопсах.
* В байтах в секунду.
* Это безразмерная величина.
* Верного ответа нет.

Вопрос.Что называется латентностью коммуникационной сети?

* Время от начала инициализации посылки до начала приёма сообщения.
* Время от окончания инициализации посылки до начала приёма сообщения.
* Время от окончания инициализации посылки до окончания приёма сообщения.
* Время от начала инициализации посылки до окончания приёма сообщения.
* Скорость передачи данных по сети.
* Верного ответа нет.

Вопрос. Что называется латентностью коммуникационной сети?

* Интервал времени от начала инициализации посылки сообщения до момента начала физической отправки по коммуникационной сети.
* Время от окончания инициализации посылки до окончания приёма сообщения.
* Время от начала инициализации посылки до начала приёма сообщения.
* Скорость передачи данных по сети.
* Время от начала инициализации посылки сообщения до окончания его приёма.
* Верного ответа нет.

Вопрос. Что понимают под топологией коммуникационной сети?

* количество узлов и связей между ними
* верного ответа нет
* конфигурацию графа сети
* латентность и пропускную способность коммуникационной сети

Вопрос. По каким параметрам топология звезда превосходит топологию кольцо?

* ни по каким
* длина критического пути
* сложность
* связность

Вопрос. Какова топология коммуникационной сети суперкомпьютера «Ломоносов»?

* Полносвязная топология
* Трёхмерный тор
* Двоичный гиперкуб
* Другая
* Толстое дерево

## Вопрос. Диффузная балансировка загрузки процессоров

* Является развитием метода коллективного решения
* Применима, если вычислительная сложность элементарных заданий априори неизвестна и непредсказуемо меняется от шага к шагу вычислений
* Приводит к росту накладных расходов обусловленному затратами на передачу данных
* Является развитием метода геометрического параллелизма
* Способствует сокращению времени выполнения вычислений на системах в которых эффективная вычислительная мощность процессоров меняется в ходе выполнения расчетов
* Направлена на сокращение простоев процессоров
* Является методом динамической балансировки загрузки
* Приводит к росту накладных расходов, обусловленному затратами на организацию управление вычислительным процессом
* Требует предварительного распределения данных между процессорами на основе принципов равенства
* объёмов вычислений и минимизации объёмов передаваемых данных между процессорами

## Часть 3. Код, OpenMP и MPI и все, все, все

### OpenMP

Вопрос. Отметьте верные утверждения о технологии OpenMP:

* Число параллельных нитей OpenMP-приложения может задаваться отдельно для каждого параллельного цикла.
* Число параллельных нитей OpenMP-приложения может задаваться до старта программы.
* Число параллельных нитей OpenMP-приложения не может задаваться пользователем.
* Число параллельных нитей OpenMP-приложения может определяться переменной окружения
* Число параллельных процессов OpenMP приложения определяется переменной окружения
* Число параллельных процессов OpenMP приложения определяется переменной окружения или задаётся явно в программе
* Все переменные программы делятся на четыре класса: локальные, общие, внешние и внутренние.
* Все переменные программы делятся на три класса: локальные, общие и внешние.
* Все переменные программы делятся на два класса: общие и внешние.
* Все переменные программы делятся на два класса: локальные и общие
* Технология OpenMP ориентирована в первую очередь на написание программ для векторных процессоров.
* Технология OpenMP ориентирована в первую очередь на написание программ для графических процессоров.
* Технология OpenMP ориентирована в первую очередь на написание программ для кластеров.
* OpenMP ориентирован в первую очередь на написание программ для векторно-конвейерных компьютеров
* OpenMP ориентирован в первую очередь на написание программ для компьютеров с

распределенной памятью

* Основная функциональность OpenMP реализуется с помощью ассемблерных вставок.
* Основная функциональность OpenMP реализуется с помощью спецкомментариев (прагм).
* Большинство конструкций OpenMP реализуется с помощью спецкомментариев
* Весь параллелизм приложения реализуется с помощью замков.
* Весь параллелизм приложения реализуется с помощью критических секций.
* Весь параллелизм приложения реализуется с помощью параллельных циклов
* Механизм параллельных секций реализует конечный параллелизм программы.
* Весь параллелизм приложения реализуется с помощью параллельных секций

Вопрос: В какой момент порождаются параллельные процессы в OpenMP:

* при входе в параллельную секцию
* при запуске программы
* при вычислении аргументов функции eval
* не порождаются
* верного ответа нет

Вопрос. В какой момент порождаются параллельные процессы в OpenMP?

* При выполнении директивы parallel.
* При входе в параллельную секцию
* (других ответов не было)

Вопрос. В какой момент уничтожаются параллельные нити в OpenMP?

* После завершения параллельной секции.
* Никогда не уничтожаются.
* Верного ответа нет.
* При выходе из критической секции.
* При входе в параллельный цикл.

Вопрос. Технология программирования OpenMP расширяет язык за счёт:

* новых директив и специальных комментариев
* новых ключевых слов
* новых библиотечных функций и переменных

Вопрос. Какие возможны варианты опции schedule директивы for в OpenMP?

* auto
* static
* single
* guided
* runtime
* section
* верных ответов нет
* private
* nowait
* dynamic

Вопрос. Что произойдёт, если все нити в программе на OpenMP вызовут директиву barrier, но из разных мест программы?

* Программа завершится немедленно с порождением исключения
* Произойдёт барьерная синхронизация
* Часть нитей будет заблокирована, а часть пойдёт выполнять программу дальше
* Все нити будут заблокированы
* Верного ответа нет

Вопрос. В чём отличие опций private и firstprivate в функциях OpenMP?

* Верного ответа нет
* Переменные из списка опции firstprivate инициализируются нулевыми значениями
* Переменные из списка опции private инициализируются значениями из нити-мастера
* Переменные из списка опции firstprivate инициализируются значениями из нити-мастера
* Отличий нет
* Переменные из списка опции private инициализируются нулевыми значениями

Вопрос. Выберите варианты директив, которые можно использовать для распределения итераций цикла между потоками:

* #pragma omp for schedule (auto, 1)
* #pragma omp for schedule (dynamic)
* #pragma omp for schedule (static, 1024)
* #pragma omp for schedule (nowait)
* #pragma omp for schedule (dynamic, guided, 3)
* #pragma omp for schedule (static, 2)
* #pragma omp for schedule (random)
* #pragma omp for schedule (dynamic, guided)
* #pragma omp for schedule (static)

Вопрос. Выберите варианты директив, которые можно использовать для распределения итераций цикла между потоками:

* #pragma omp for, если данная директива находится внутри параллельного фрагмента.
* #pragma parallel for
* #pragma omp parallel
* #pragma omp parallel for
* #pragma for

Вопрос. Какими нитями будет выполнена критическая секция в OpenMP?

* Верного ответа нет
* Всеми нитями параллельной области
* Ни одной нитью
* Только нитью-мастером
* Только одной произвольной нитью

### 

### MPI

Вопрос. Отметьте верные утверждения:

* MPI - это сокращение от My Personal Identifier.
* MPI - это сокращение от Message Passing Interface.
* MPI - это сокращение от Multiple Parallel Interface.
* MPI – это сокращение от Master Parallel Instructions.
* SPMD - это сокращение от Single Program Multiple Data.
* MPI не поддерживает SPMD-модель параллельного программирования
* Каждый параллельный процесс в MPI имеет номер.
* Параллельный процесс в MPI в каждой группе имеет уникальный номер (ранг).
* Параллельный процесс в MPI может иметь только один номер (ранг).
* Каждый параллельный процесс в MPI может иметь несколько номеров.
* Использовать функции MPI можно только после вызова MPI\_Init.
* Использовать большинство функций MPI можно только после вызова MPI\_Init.
* Использовать большинство функций MPI можно только после вызова MPI\_Finalize.
* Использовать большинство функций MPI можно только до вызова MPI\_Finalize.
* Функция MPI\_Comm\_size определяет общее число запущенных параллельных процессов приложения.
* Функция MPI\_Comm\_size определяет число параллельных процессов коммуникатора
* Функция MPI\_Comm\_rank определяет число параллельных процессов коммуникатора.
* Функция MPI\_Comm\_rank определяет общее число запущенных параллельных процессов приложения.
* Функция MPI\_Comm\_rank определяет число линейно независимых строк матрицы инцидентности

параллельных процессов.

\*MPI\_Comm\_rank определяет номер процесса в коммуникаторе\*

* Использование функций MPI\_Isend и MPI\_Irecv не может привести к тупиковой ситуации (deadlock).
* Использование функций MPI\_Send и MPI\_Recv может привести к тупиковой ситуации (deadlock)
* На MPI можно реализовать схему взаимодействия каждого процесса с каждым.
* На MPI невозможно реализовать схему взаимодействия «мастер-рабочие».
* Функция, соответствующая коллективной операции, должна быть вызвана только двумя процессами.
* Функция, соответствующая коллективной операции, должна быть вызвана только одним процессом.
* Функция, соответствующая коллективной операции, должна быть вызвана только процессом-мастером
* Функция, соответствующая коллективной операции, должна быть вызвана каждым процессом, быть может, со своим набором параметров (не сказано про то, что они должны быть вызваны в рамках коммуниактора)
* MPI\_Bsend – это пример коллективной операции.
* MPI\_Barrier - это пример коллективной операции
* MPI\_Bcast – это пример коллективной операции.
* MPI\_Isend - это пример коллективной операции возврат процесса из функции, реализующей коллективную операцию, всегда означает, что операция уже завершена
* В коллективных операциях участвуют все процессы некоторого коммуникатора
* В коллективных операциях участвуют все процессы приложения, кроме процесса с номером 0.
* В коллективных операциях участвуют все процессы приложения
* Возврат процесса из функции, реализующей коллективную операцию, всегда означает, что операция уже завершена.
* Возврат процесса из функции, реализующей коллективную операцию, не обязательно означает, что операция уже завершена.
* Возврат процесса из функции, реализующей коллективную операцию, не означает, что операция уже завершена

Вопрос. Отметьте верные утверждения о функции MPI\_Recv:

* перед вызовом функции MPI\_Recv надо обратиться к функции MPI\_Get\_count;
* функцией MPI\_Recv нельзя принимать сообщение, посланное с помощью функции

MPI\_Ssend;

* нельзя использовать функцию MPI\_Recv, если мы не знаем заранее отправителя сообщения

или тег сообщения.

* возврат из функции означает, что либо произошла ошибка, либо принятое сообщение

расположено в первом параметре;

Вопрос. Отметьте верные утверждения о тупиковых ситуациях в MPI:

* Функции MPI\_Isend и MPI\_Irecv можно использовать для гарантии невозникновения тупиковой ситуации (deadlock)
* Использование функций MPI\_Ssend и MPI\_Recv не может привести к тупиковой ситуации (deadlock)
* Функция MPI\_Sendrecv является блокирующей, но не приводит к тупиковой ситуации (deadlock)
* Использование функций MPI\_Send и MPI\_Recv может привести к тупиковой ситуации (deadlock)

Вопрос. Буферизованная посылка сообщения с блокировкой MPI\_Bsend означает, что возврат из функции произойдёт тогда, когда:

* Сообщение будет скопировано из буфера посылки в специально выделенный буфер.
* Сообщение покинет процессор.
* Адресат инициировал приём данного сообщения.
* Никогда.
* Верного ответа нет.
* Сообщение покинет процесс.
* Сообщение принято адресатом.

Вопрос: Посылка сообщения с блокировкой (MPI\_Send) означает, что возврат из функции произойдёт тогда, когда:

* Сообщение принято адресатом;
* Сообщение покинет процесс
* Можно повторно использовать параметры данной функции;
* Адресат инициировал приём данного сообщения.
* Сообщение покинет процессор;
* Данные скопированы из буфера посылки;

Вопрос. Возврат из функции MPI\_Wait означает:

* Завершение выполнения неблокирующей операции.
* Начало выполнения неблокирующей операции.
* Возможность повторного использования буфера посылки.
* Инициализацию неблокирующей операции, но ничего не говорит о начале или завершённости обмена.
* Верного ответа нет.

Вопрос. Процессы параллельной программы в рамках MPI:

* обязательно выполняются на одном процессоре.
* могут выполняться на разных процессорах, на одном процессоре могут располагаться несколько

процессов,

* могут выполняться только на разных процессорах,

Вопрос. Под параллельной программой в рамках MPI понимается:

* множество одновременно работающих процессоров.
* множество одновременно выполняемых процессов,
* множество одновременно выполняемых потоков,

Вопрос. Топология типа тор в MPI является частным видом топологии типа:

* декартовой топологии,
* полного графа.
* графа произвольного вида

Вопрос. Возврат из неблокирующей операции в MPI означает

* Инициализацию запрошенной операции, но ничего не говорит о начале или завершённости обмена.
* Возможность повторного использования буфера посылки данной функции.
* Завершение пересылки данных.
* Начало пересылки данных.

Вопрос. Если size – число процессов в каком-то коммуникаторе MPI, то в каком диапазоне лежит номер процесса, возвращаемый функцией MPI\_Comm\_rank?

* От 0 до 1
* От 0 до size-1
* От 1 до size-1
* От 0 до size
* Верного ответа нет
* От 1 до size

Вопрос. Что произойдёт, если процессы некоторого коммуникатора вызовут функцию MPI\_Barrier из разных мест программы?

* Верного ответа нет
* Произойдёт барьерная синхронизация
* Программа завершится немедленно с порождением исключения
* Часть процессов коммуникатора будет заблокирована, а часть пойдёт выполнять программу дальше
* Все процессы коммуникатора будут заблокированы

Вопрос. В чём основная особенность функции MPI\_Alltoallw?

* Верного ответа нет
* Возможна посылка данных процессам из разных коммуникаторов
* Возможна посылка нескольких порций данных каждому процессу коммуникатора
* Возможна посылка данных разных типов
* Возможна посылка данных не всем процессам коммуникатора

Вопрос. Какие наборы номеров процессов в MPI могут задать группу?

* никакие
* -1,0,1
* 0,2,5
* 5
* 2,2,0,5

Вопрос. Фрагмент, приведённый ниже:

*MPI\_Comm\_size( comm, &size);*

*MPI\_Send( buf, 15, MPI\_INT, size+1, 8, comm); //в comm всего size процессов*

*MPI\_Send( buf, 8, MPI\_INT, size-1, 15, comm);*

* заведомо содержит две ошибки;
* заведомо содержит четыре ошибки;
* заведомо содержит одну ошибку;
* заведомо содержит три ошибки;
* может быть правильным;

Вопрос. Фрагмент, приведённый ниже:

*MPI\_Comm\_size( comm, &size);*

*MPI\_Send( buf, 15, MPI\_INT, size+1, 8, comm); //в comm всего size процессов*

*MPI\_Send( buf, 8, MPI\_INT, size-1, 15, comm);*

*int MPI\_Comm\_size ( MPI\_Comm comm, int \*size );*

*int MPI\_Send( void \*buf, int count, MPI\_Datatype datatype, int dest, int tag, MPI\_Comm comm );*

* может быть правильным;
* заведомо содержит одну ошибку;
* заведомо содержит три ошибки;
* заведомо содержит четыре ошибки;
* заведомо содержит две ошибки;"

### 

### Linda

Вопрос: Как порождаются параллельные процессы в системе Linda

* при запуске программы
* при вычислении аргументов функции eval
* никак
* верного ответа нет
* при входе в параллельную секцию

Вопрос. Что произойдет, если в системе Linda занести один и тот же кортеж в пространство кортежей (ПК) дважды?

* в ПК будет две копии этого кортежа
* кортеж занесётся в ПК дважды с разными именами
* в ПК останется только одна копия этого кортежа
* в ПК не останется ни одной копии этого кортежа выдастся сообщение об ошибке

Вопрос. Что произойдёт, если в системе Linda при считывании кортежа из пространства кортежей под маску подходит несколько кортежей?

* выберется случайный кортеж
* выберутся все подходящие кортежи
* выдастся сообщение об ошибке
* выберется первый по времени занесения кортеж
* выберется последний по времени занесения кортеж

Вопрос. Какие структуры данных, кроме кортежей, дополнительно вводятся в системе Linda?

* структуры
* никаких
* списки
* массивы
* какие-то другие

### 

### CUDA C

Вопрос. Что означает квалификатор функции \_\_global\_\_ в языке программирования CUDA C?

* Функция является ядром, т.е. исполняется на ГПУ, а запускается с хоста (в некоторых версиях — также и с ГПУ) особым образом.
* других вариантов не было

Вопрос. Что означает квалификатор функции \_\_device\_\_ в языке программирования CUDA C?

* Функция может исполняться только на ГПУ.
* Функция является ядром, т.е. исполняется на ГПУ, а запускается с хоста (в некоторых версиях — также и с ГПУ) особым образом.
* функция может исполняться на хосте и на ГПУ.
* Верного ответа нет.
* Функция может исполняться только на хосте.

### 

### Остальное

Вопрос. В чём основная идея распараллеливающего компилятора?

* Параллелизм программ не используется
* Верных ответов нет
* Компилятор может извлекать и использовать параллелизм программ
* Аппаратура может извлекать и использовать параллелизм программ
* Программист должен помогать компилятору извлекать и использовать параллелизм программ

Вопрос. Какой смысл вкладывается в понятие эквивалентности, когда мы говорим об эквивалентных преобразованиях программ?

* Равенство объёмов занимаемой памяти.
* Полное совпадение конечного результата работы программ.
* Идентичность текстов программ.
* Равенство времени выполнения программ.
* Результаты выполнения могут отличаться из-за ошибок округления.

Вопрос. Какой смысл вкладывается в понятие эквивалентности, когда мы говорим об эквивалентных преобразованиях программ?

* Результаты выполнения могут отличаться из-за ошибок округления.
* Равенство объёмов занимаемой памяти.
* Идентичность текстов программ.
* Верного ответа нет.
* Равенство времени выполнения программ.

Вопрос. Отметьте циклы на языке Си, которые можно векторизовать при наличии соответствующих векторных команд:

* for(i=0; i <n; i++) b[i]=a[i]+s;
* for(i=0; i<n; i++) c[i]=a[i]+b[i];
* for(i=0; i<n; i++) a[i]="a[i-1]+b[i];

Вопрос. Отметьте верные утверждения о векторной обработке:

* Диагональ квадратной матрицы может быть вектором
* Аргументом векторной операции может быть скаляр
* Наличие истинных информационных зависимостей не позволяет векторизовать цикл
* В современных микропроцессорах векторная обработка не используется
* Под векторизацией программы подразумевается замена всех переменных программы векторами

## 

## 

## Часть 4. Графы алгоритмов

Представим программу в виде ориентированного графа. В качестве вершин могут выступать процедуры, циклы, линейные участки, операторы, итерации циклов, срабатывания операторов. Дуги отражают отношения двух типов: операционное и информационное.

Операционное отношение - вершины A и B соединяются дугой (от A к B) тогда и только тогда, когда вершина B может быть выполнена сразу после вершины A. (отношение по передаче управления. Показывает, как вершины *будут* исполняться программой).

Информационное отношение - вершины A и B соединяются дугой (от A к B) тогда и только тогда, когда вершина B использует в качестве аргумента некоторое значение, полученное в вершине A. (отношение по передаче данных. Показывает, как вершины *могут* исполняться программой, чтобы результат остался тем же).

Граф управления: вершины - операторы, дуги - операционное отношение. (коротенький)

Информационный граф: вершины - операторы, дуги - информационное отношение. (коротенький)

Операционная история: вершины - срабатывания операторов, дуги - операционное отношение. (длинненький). Всегда есть ровно одна начальная вершина без входящей дуги, ровно одна конечная вершина без выходящей дуги, у всех остальных вершин есть ровно одна входящая дуга и одна исходящая.

Информационная история: вершины - срабатывания операторов, дуги - информационное отношение. (длинненький). Ацикличенский, без кратных дуг, поэтому число дуг не превосходит n(n-1)/2, где n - число вершин.

Вопрос. В каких графовых моделях количество вершин не зависит от значений входных переменных:

* граф управления
* во всех моделях всегда зависит
* информационный граф
* информационная история
* операционная история
* во всех моделях никогда не зависит

### 

### 

### Может ли

Вопрос. Для некоторого фрагмента программы построили четыре модели. Может ли оказаться несвязным графом:

* информационный граф
* информационная история
* операционная история Нет (в книге написано, что операционно-логической историей называется

ориентированный граф, который представляет из себя единственный путь от начальной вершины к

конечной)

граф управления (пример с goto в лекции 13)

Вопрос. Может ли управляющий граф некоторого фрагмента программы быть связным, а его информационный граф несвязным?

* Только для фрагментов, в которых нет циклов
* Нет
* Да
* Управляющий граф не может быть несвязным Это верно (хоть и не имеет отношения к делу)
* Только для фрагментов, записанных на языке Си Нет

Вопрос. Может ли информационная история некоторого фрагмента содержать 5 вершин и 9 дуг?

* нет
* да

Вопрос. Может ли информационная история некоторого фрагмента содержать 7 вершин и 11 дуг?

* да
* нет

Вопрос. Может ли информационная история некоторого фрагмента содержать 102 вершины и лишь 7 дуг?

* нет
* да

### 

### Верно ли

ЯПФ - ярусно-параллельная форма. Сторится по информационной истории и задается двумя правилами:

1. Начальная вершина каждой дуги расположена на ярусе с номером меньшим, чем номер яруса конечной вершины.
2. Между вершинами, расположенными на одном ярусе, не может быть дуг.

Число ярусов - число шагов, необходимых для параллельного исполнения программы (при достаточном количестве процессоров).

Вопрос. Отметьте правильные утверждения, касающиеся ЯПФ графа

* Структура графа определяет собой ограничения, дающие большую или меньшую свободу для разбиения на ярусы
* ЯПФ графа однозначно определяется структурой графа
* При разбиении получающиеся подмножества должны быть непересекающимися (если имеются в виду ярусы)
* Для определения ЯПФ графа на некие подмножества разбивается множество вершин графа
* Для определения ЯПФ графа на некие подмножества разбиваются множества вершин и дуг графа
* При разбиении получающиеся подмножества могут быть как пересекающимися, так и непересекающимися
* Не существует графов, у которых ЯПФ (если не считать возможными пустые ярусы) единственна
* Существуют графы, у которых ЯПФ (если не считать возможными пустые ярусы) единственна (граф из двух вершин, соединенных дугой)
* ЯПФ графа не зависит от структуры графа
* Для определения ЯПФ графа на некие подмножества разбивается множество дуг графа
* При разбиении получающиеся подмножества должны быть пересекающимися

Вопрос.Отметьте, какие утверждения об информационной истории (ИИ) программы верны:

* Число вершин в ИИ не зависит от входных данных программы.
* ИИ фрагмента программы не может совпадать с его операционной историей.
* Существуют программы, ИИ которых нельзя построить с помощью статического анализа.
* ИИ - это линейный ориентированный граф, каждая вершина которого имеет не более одной входной

дуги и не более одной выходной дуги. (это верно для операционной истории)

* ИИ любой программы можно построить с помощью статического анализа.
* ИИ фрагмента программы может совпадать с его операционной историей.

Вопрос. Отметьте, какие утверждения об операционной истории (ОИ) программы верны:

* Число вершин в ОИ зависит от входных данных программы . Не зависит??? Зависит. Например, цикл длины n
* ОИ любой программы можно построить с помощью статического анализа
* ОИ - это линейный ориентированный граф, каждая вершина которого имеет не более одной

входной дуги и не более одной выходной дуги

* ОИ фрагмента программы не может совпадать с его информационной историей
* ОИ фрагмента программы не может совпадать с его графом управления

Вопрос. Отметьте, какие утверждения об операционной истории (ОИ) программы верны:

* Число вершин в ОИ зависит от входных данных программы
* ОИ любой программы можно построить с помощью статического анализа
* ОИ - линейный ориентированный граф, каждая вершина которого может иметь любое число входных и выходных дуг
* ОИ фрагмента программы может совпадать с его информационной историей

Вопрос. Верно ли, что информационная история любого фрагмента всегда содержит больше дуг, чем граф управления того же фрагмента?

* нет
* да

Вопрос. Верно ли, что информационная история любого фрагмента всегда содержит больше вершин, чем информационный граф того же фрагмента?

* нет
* да

Вопрос. Верно ли, что информационная история может содержать больше вершин, чем информационный граф?

* нет
* да

Вопрос. Верно ли, что в информационной истории фрагмента программы число дуг не может превосходить числа вершин более, чем в пять раз?

* нет
* да

Вопрос. Верно ли, что в информационной истории фрагмента программы число дуг не может превосходить числа вершин более, чем в пять(три) раз?

* да
* нет

## Непонятная хрень

Вопрос. Сортировка данных. Отметьте верные утверждения

* При больших объёмах сортируемых данных алгоритм сортировки слиянием менее чувствителен к размеру кэш памяти, чем алгоритм пирамидальной сортировки
* блоки данных размещенные на каждом из процессоров при упорядочивании сетью сортировки со слиянием Бэтчера должны быть одинакового размера
* Известен алгоритм построения минимальных по числу шагов сетей сортировки для любого числа процессоров
* Последовательность выполнения операций компараторов слияния в сети сортировки не зависит от упорядоченности элементов исходного массива
* Из того, что оценки времён выполнения двух алгоритмов A1 и A2 находятся в отношении O(A1)
* В общем случае сети сортировки эффективны при выполнении на одном процессоре
* Объем данных сортируемых с помощью алгоритма сдваивания ограничен объемом оперативной памяти одного вычислительного узла
* Сети сортировки обеспечивают возможность построения эффективных параллельных алгоритмов сортировки

Вопрос. Иерархический алгоритм рациональной декомпозиции графов предполагает:

* Этап первичного огрубления графа без потерь данных
* Этап первичного огрубления графа с потерей данных
* Наличие информации о координатах узлов сетки
* этап локального уточнения, на котором узлы перераспределяются между соседними доменами
* Наличие информации о предварительном распределении узлов сетки по доменам
* Этап записи в файл результатов декомпозиции

Вопрос: Критерии декомпозиции сеточных графов при решении задач на неадаптивных сетках на кластерных вычислительных системах включают в себя: (нет ответа)

* связность множества размещённых в каждом из доменов элементов графа
* равномерность распределения элементов графа по доменам
* Близость геометрических координат узлов, размещённых в одном домене
* минимальность времени выполнения декомпозиции
* минимальность максимального числа ребер, соединяющих вершины каждого из доменов с вершинами других доменов
* минимальность числа смежных доменов

### 