

34. Последовательная и параллельная сложность алгоритмов.

Последовательная сложность алгоритма – число операций, которые нужно выполнить при его последовательном выполнении

Параллельная сложность алгоритма – число шагов, за которое можно выполнить алгоритм в предположении неограниченного числа необходимых процессоров (ядер, вычислительных узлов и т.д.)

Граф алгоритма — это ориентированный ациклический мультиграф, вершины которого соответствуют операциям алгоритма, а дуги — передаче данных между ними. Вершины графа алгоритма могут соединяться несколькими дугами, в частности, когда в качестве разных аргументов одной и той же операции используется одна и та же величина. Граф алгоритма почти всегда является параметризованным графом. Он используется как удобное представление алгоритма при исследовании его структуры, ресурса параллелизма, а также других свойств. Его можно рассматривать как параметризованную информационную историю. Он сохраняет ее информативность, при этом обладая компактностью за счет параметризации. Граф можно построить по исходному тексту программы.

Ярусно параллельная форма (ЯПФ) – представление графа алгоритма, в котором:

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

Высота ЯПФ – число ярусов. **Ширина яруса** – число вершин, содержащихся в ярусе. **Ширина ЯПФ** – максимальная ширина ярусов в ЯПФ.

Каноническая ЯПФ – ЯПФ, высота которой на единицу больше длины критического пути, а входные вершины расположены в первом ярусе. Для заданного графа его каноническая ЯПФ единственна. Высота канонической ЯПФ соответствует параллельной сложности алгоритма.

Критический путь графа – путь максимальной длины в ориентированном ациклическом графе.

Ресурс параллелизма – это где и насколько мы можем сделать программу более эффективной за счет распараллеливания.