
Понятие алгоритма. Свойства алгоритмов. Показатели эффективности алгоритмов. Анализ времени выполнения алгоритмов.

Алгоритм (algorithm) – это конечная последовательность инструкций исполнителю, в результате выполнения которых обеспечивается получение из входных данных требуемого выходного результата (решение задачи).

Свойства алгоритма:

- **Дискретность** – алгоритм представляется как последовательность инструкций исполнителя. Каждая инструкция выполняется только после того, как закончилось выполнение предыдущей команды.*
 - **Конечность** (результативность, финитность) – алгоритм должен заканчиваться после выполнения конечного числа инструкций.
 - **Детерминированность** – каждый шаг алгоритма должен быть однозначно определен – записан на формальном языке исполнителя. Детерминированность обеспечивает совпадение результатов, получаемых при многократном выполнении алгоритма, на одном и том же наборе входных данных.
 - **Массовость** – алгоритм решения задачи должен быть применим для некоторого класса задач, различающихся лишь значениями входных данных.
-

Показатели эффективности алгоритма

- **Количество операций** – временная эффективность (time efficiency), показывает насколько быстро работает алгоритм.
 - **Объем потребляемой памяти** – пространственная эффективность (space efficiency), отражает максимальное количество памяти, требуемой для выполнения алгоритма.
-

Показатели эффективности позволяют:

- *Оценить потребности алгоритма в вычислительных ресурсах: процессорном времени, памяти, пропускной способности сети*
- *Сравнивать алгоритмы между собой*

(Существуют также другие показатели, которые имеет смысл рассматривать, если они значительно влияют на процесс выполнения задачи.)

Анализ времени выполнения алгоритмов:

На время выполнения алгоритма влияют

1. *Размер входных данных*
2. *Качество реализации алгоритма на языке программирования*
3. *Качество скомпилированного кода*
4. *Производительность вычислительной машины*

Для большинства алгоритмов количество выполняемых ими операций напрямую зависит от размера входных данных. Например, в алгоритме поиска наибольшего элемента время выполнения определяется не значениями в массиве, а его длиной n
