

Ответ на вопрос  
по дисирации:

## O. Сложность алгоритмов

Big O - верхняя граница (наихудший случай), временной сложности алгоритма ("order of")

1) рост алгоритма от роста входных данных

Примеры нотаций:

а)  $O(1)$  - константная нотация - время алгоритма не зависит от входных данных.

Пример: Поиск доступа к элементу массива по элементу

б)  $O(\log n)$  - логарифмическая сложность - выполнение алгоритма растет медленно с увеличением размера входных данных

Пример: Бинарный поиск

в)  $O(n)$ : Лин. сложность - выполнение алгоритма пропорционально размеру входных данных.

Пример: Пресмотр всех элементов в массиве

г)  $O(n \log n)$ : Линейно-логарифмическая сложность:  
 $O(n) \leq O(n \log n) \leq O(n^2)$

Пример: Сортировка слиянием

д)  $O(n^2)$ : Квадратичная сложность - квадрат размера входных данных.

Пример: Bubble sort

е)  $O(n^3)$  Кубическая сложность

Пример: Метод множественной обработки данных



н.)  $O(n!)$  Факториальная сложность  
Перебор всех возможных вариантов, неэффективный  
для больших значений  $n$ .

Другие обозначения сложности алгоритмов:

- Big Theta ( $\Theta$ ) - верхние границы и нижние границы временной сложности;
- Big Omega ( $\Omega$ ) - нижняя граница времен. сложности;
- Little O ( $o$ ): верхняя граница, которая строже, чем Big O;
- Little ( $\omega$ ): нижняя граница, которая строже, чем Big Omega;

