Асимптотические обозначения

1. Точная оценка.

```
INSERTION-SORT(A)
for j = 2 to A.length
    key = A[j]
    // Вставка A[j] в отсортированную
        последовательность A[1..j − 1].
    i = j − 1
    while i > 0 и A[i] > key
        A[i + 1] = A[i]
        i = i − 1
    A[i + 1] = key
```

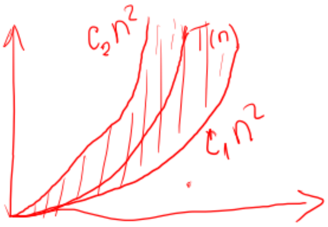
$T(n)$ — время работы алгоритма размера вход. данных

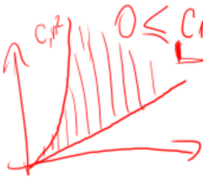
$T(n) = \Theta(h(n))$

$n,\ n^2,\ n\log n,\ n^3$

$$\Theta(h(n)) = \begin{cases} T(n) : \exists\, c_1>0,\ c_2>0,\ n_0>0 \\ \forall n>n_0 : \end{cases}$$

$$0 \le c_1 h(n) \le T(n) \le c_2 h(n)$$



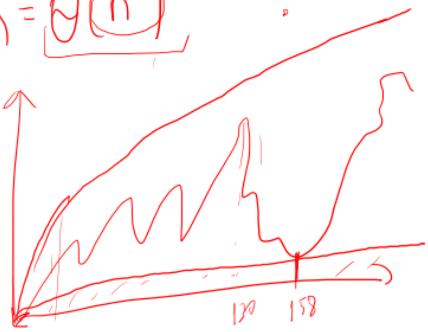


$T(n) = \Theta(n^2)$

2. Верхняя оценка.

$$O(h(n)) = \begin{cases} T(n),\ \exists\, c>0,\ n_0>0 \\ \forall n \ge n_0 \quad 0 \le T(n) \le c\cdot h(n) \end{cases}$$

$h(n)$ - верхняя асимптотич. оценка



3. Нижняя оценка.

3. Нижняя оценка.

$$\Omega(h(n)) = \begin{cases} T(n) : \quad \exists\, c>0,\ n_0>0 \\ \forall n > n_0 \quad 0 \le c\,h(n) \le T(n) \end{cases}$$

4. Теорема о связи.

$\Theta,\ \Omega,\ O$

$$T(n) = \Theta(g(n)) \Leftrightarrow \begin{cases} T(n) = O(g(n)) \\ T(n) = \Omega(g(n)) \end{cases}$$

1) Правило сумм $T_1(n)+T_2(n) = O(\max(g_1(n), g_2(n)))$

2) Правило произведений $T_1(n)\cdot T_2(n) = O(g_1(n)\cdot g_2(n))$

3) Умножение на константу $c\,T_1(n) = O(g_1(n))$

4) Прибавление константы $T_1(n)+c = O(g_1(n))$

Док-во

$T_1(n) = O(g_1(n)) \Rightarrow \exists\, c_1, n_1 : \forall n>n_1 \quad T_1(n) \le c_1\cdot g_1(n)$ по опр

$T_2(n) = O(g_2(n)) \Rightarrow \exists\, c_2, n_2 : \forall n>n_2 \quad T_2(n) \le c_2\cdot g_2(n)$

Надо найти c_3 и n_3

$\forall n>n_3 \quad (T_1+T_2)(n) \le c_3\cdot \max(g_1(n), g_2(n))$

$\forall n_3 > \max(n_1, n_2) \quad T_1(n)+T_2(n) \le c_1 g_1(n) + c_2 g_2(n) \le$

$\le c_1\cdot \max(g_1(n), g_2(n)) + c_2\cdot \max(g_1(n), g_2(n)) \le$

$\le \underbrace{(c_1+c_2)}_{c_3} \max(g_1(n), g_2(n))$

$n_3 = \max(n_1, n_2) \qquad c_3 = c_1 + c_2$

Опр Инвариант цикла — истинное утверждение, которое сохраняется перед и после каждой итерации цикла алгоритма

Инициализация — верно до начала работы

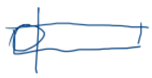
Сохранение — верно после каждой итерации

Завершение — верно после выполнения алгоритма

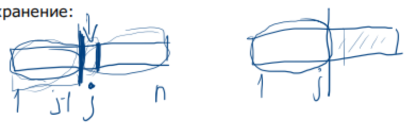
Инвариант: Массив A[1...j-1] содержит те же элементы, что были изначально, но в отсортированном виде

Инициализация:

$j=2 \quad A[1, .. j-1] = A[1]$



Сохранение:



Завершение:

$j = n+1$