

Analiza złożoności czasowej

Przykład analizy czasu

Sortowanie przez
wstawianie
(Insertion-sort)

```
InsertionSort(A)
  n = A.length
  for j=2 to n
    // A[1] ≤ A[2] ≤ ... ≤ A[j-1]
    // wstawiamy A[j]
    pom=A[j]
    i=j-1
    while i ≥ 1 and A[i] > pom
      A[i+1]=A[i]
      i=i-1
    A[i+1]=pom
```

InsertionSort(A)

n = A.length

C_1

for j=2 to n

// $A[1] \leq A[2] \leq \dots \leq A[j-1]$

// wstawiamy A[j]

pom=A[j]

i=j-1

while i>=1 and A[i]>pom

A[i+1]=A[i]

i=i-1

A[i+1]=pom

$t_j \cdot C_2$

C_3

t_j = ilość wykonań ciała pętli “while”
dla danego j

$$T(n) = c_1 + c_2(t_2 + t_3 + \dots + t_n) + c_3n$$

$$T(n) = c_1 + c_2(t_2 + t_3 + \dots + t_n) + c_3n$$

pesymistycznie:

$$\begin{aligned} T(n) &= c_1 + c_2(1 + 2 + \dots + (n-1)) + c_3n = \\ &= c_1 + c_2n(n-1) \cdot \frac{1}{2} + c_3n = \\ &= c_1 + \frac{1}{2} \cdot c_2n^2 - \frac{1}{2} \cdot c_2n + c_3n = \\ &= \frac{1}{2} \cdot c_2n^2 + (c_3 - \frac{1}{2} \cdot c_2) \cdot n + c_1 \end{aligned}$$

szacowanie asymptotyczne
złożoności czasowej

$$T(n) = 10n^2 + 1000n + 10000$$

$$n=10 : \quad 10^3 \quad 10^4 \quad 10^4$$

$$T(n) = 10n^2 + 1000n + 10000$$

$$n=10 : \quad 10^3 \quad 10^4 \quad 10^4$$

$$n=10^2 : \quad 10^5 \quad 10^5 \quad 10^4$$

$$T(n) = 10n^2 + 1000n + 10000$$

$$n=10 : \quad 10^3 \quad 10^4 \quad 10^4$$

$$n=10^2 : \quad 10^5 \quad 10^5 \quad 10^4$$

$$n=10^3 : \quad 10^7 \quad 10^6 \quad 10^4$$

$$T(n) = 10n^2 + 1000n + 10000$$

$$n=10 : 10^3 \quad 10^4 \quad 10^4$$

$$n=10^2 : 10^5 \quad 10^5 \quad 10^4$$

$$n=10^3 : 10^7 \quad 10^6 \quad 10^4$$

$$n=10^4 : 10^9 \quad 10^7 \quad 10^4$$

$$n=10^6 : 10^{13} \quad 10^9 \quad 10^4$$

$$T_1(n) = n^2$$

$$T_2(n) = 100n$$

$$n = 10$$

$$T_1(n) = 10^2$$

$$T_2(n) = 10^3$$

$$T_1(n) = n^2$$

$$T_2(n) = 100n$$

$$n = \quad 10 \quad 10^2$$

$$T_1(n) = 10^2 \quad 10^4$$

$$T_2(n) = 10^3 \quad 10^4$$

$$T_1(n) = n^2$$

$$T_2(n) = 100n$$

$$n = \quad 10 \quad 10^2 \quad 10^3 \quad 10^4 \quad 10^5$$

$$T_1(n) = 10^2 \quad 10^4 \quad 10^6 \quad 10^8 \quad 10^{10}$$

$$T_2(n) = 10^3 \quad 10^4 \quad 10^5 \quad 10^6 \quad 10^7$$

$T(n) = O(f(n))$ jeżeli istnieją stałe $c > 0$ i $n_0 > 0$ takie, że $T(n) \leq c \cdot f(n)$ dla $n > n_0$

$T(n) = \Omega(f(n))$ jeżeli istnieją stałe $c > 0$ i $n_0 > 0$ takie, że $T(n) \geq c \cdot f(n)$ dla $n > n_0$

$T(n) = \Theta(f(n))$ jeżeli $T(n) = O(f(n))$ oraz $T(n) = \Omega(f(n))$ (czyli jeżeli istnieją stałe $c_1 > 0$, $c_2 > 0$ i $n_0 > 0$ takie, że $c_1 \cdot f(n) \leq T(n) \leq c_2 \cdot f(n)$ dla $n > n_0$)

$T(n) = O(f(n))$ jeżeli $T(n) \leq c \cdot f(n)$ dla jakiejś stałej c , dla wszystkich n (poza pewną ilością początkowych n)

$T(n) = \Omega(f(n))$ jeżeli $T(n) \geq c \cdot f(n)$ dla jakiejś stałej c , dla wszystkich n (poza pewną ilością początkowych n)

$T(n) = \Theta(f(n))$ jeżeli $c_1 \cdot f(n) \leq T(n) \leq c_2 \cdot f(n)$ dla jakichś stałych c_1, c_2 , dla wszystkich n (poza pewną ilością początkowych n)

$$T_1(n) = n^2$$

$$T_2(n) = 100n$$

$$n = \quad 10 \quad 10^2 \quad 10^3 \quad 10^4 \quad 10^5$$

$$T_1(n) = 10^2 \quad 10^4 \quad 10^6 \quad 10^8 \quad 10^{10}$$

$$T_2(n) = 10^3 \quad 10^4 \quad 10^5 \quad 10^6 \quad 10^7$$

$$T_1(n) = \Theta(n^2)$$

$$T_2(n) = \Theta(n)$$

$$T(n) = 10n^2 + 1000n + 10000$$

| | | | |
|------------|-----------|--------|--------|
| $n=10 :$ | 10^3 | 10^4 | 10^4 |
| $n=10^2 :$ | 10^5 | 10^5 | 10^4 |
| $n=10^3 :$ | 10^7 | 10^6 | 10^4 |
| $n=10^4 :$ | 10^9 | 10^7 | 10^4 |
| $n=10^6 :$ | 10^{13} | 10^9 | 10^4 |

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

$$T(n) = c_1 + c_2(t_2 + t_3 + \dots + t_n) + c_3n$$

pesymistycznie:

$$\begin{aligned} T(n) &= c_1 + c_2(1 + 2 + \dots + (n-1)) + c_3n = \\ &= c_1 + c_2n(n-1) \cdot \frac{1}{2} + c_3n = \\ &= c_1 + \frac{1}{2} \cdot c_2n^2 - \frac{1}{2} \cdot c_2n + c_3n = \\ &= \frac{1}{2} \cdot c_2n^2 + (c_3 - \frac{1}{2} \cdot c_2) \cdot n + c_1 \\ &= \Theta(n^2) \end{aligned}$$

czyli pesymistyczny czas działania
sortowania przez wstawianie jest $\Theta(n^2)$

czas pesymistyczny

$T(d)$ - czas działania pewnego algorytmu
na danych d

$T_p(n)$ - czas pesymistyczny tego algorytmu
na danych rozmiaru n

$$T_p(n) = \max\{T(d) : |d| = n\}$$

$T_p(n)$ można szacować z góry: $T_p(n) = O(f(n))$,
z dołu: $T_p(n) = \Omega(f(n))$, czy z obu stron: $T_p(n) = \Theta(f(n))$

czas optymistyczny

$T(d)$ - czas działania pewnego algorytmu
na danych d

$T_o(n)$ - czas optymistyczny tego algorytmu
na danych rozmiaru n :

$$T_o(n) = \min\{T(d) : |d| = n\}$$

$T_o(n)$ można szacować z góry: $T_o(n) = O(f(n))$,
z dołu: $T_o(n) = \Omega(f(n))$, czy z obu stron: $T_o(n) = \Theta(f(n))$

Stwierdzenie. Pesymistyczny czas sortowania kopcowego jest $\Theta(n \cdot \lg n)$, gdzie n to rozmiar sortowanej tablicy.

Dowód.

...