# Analiza złożoności czasowej

interesuje nas jak zmienia się czas działania algorytmu w zależności od rozmiaru danych

jednostki czasu są umowne

# 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
  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
```

$c_1$

$c_3$

$t_j \cdot c_2$

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

$$T(n) = c_1 + c_2(t_2 + t_3 + \ldots + t_n) + c_3(n-1)$$

$$T(n) = c_1 + c_2(t_2+t_3+\ldots+t_n) + c_3(n-1)$$

pesymistycznie:

$$T(n) = c_1 + c_2(1+2+\ldots+(n-1)) + c_3(n-1) =$$

$$= c_1 + c_2\, n(n-1)\cdot\tfrac{1}{2} + c_3(n-1) =$$

$$= c_1 + \tfrac{1}{2}\cdot c_2\, n^2 - \tfrac{1}{2}\cdot c_2\, n + c_3(n-1) =$$

$$= \tfrac{1}{2}\cdot c_2\, n^2 + (c_3 - \tfrac{1}{2}\cdot c_2)\cdot n + c_1 - c_3$$

# szacowanie asymptotyczne złożoności czasowej

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

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

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

| | | | |
|---|---|---|---|
| $n=10$ : | $10^3$ | $10^4$ | $10^4$ |
| $n=10^2$: | $10^5$ | $10^5$ | $10^4$ |

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

| | | | |
|---|---|---|---|
| $n=10$ : | $10^3$ | $10^4$ | $10^4$ |
| $n=10^2$: | $10^5$ | $10^5$ | $10^4$ |
| $n=10^3$: | $10^7$ | $10^6$ | $10^4$ |

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

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

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

$$T_2(n) = 100n$$

$$n = \quad 10$$

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

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

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

$$T_2(n) = 100n$$

| | | |
|---|---|---|
| $n =$ | $10$ | $10^2$ |
| $T_1(n) =$ | $10^2$ | $10^4$ |
| $T_2(n) =$ | $10^3$ | $10^4$ |

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

$$T_2(n) = 100n$$

| | | | | | |
|---|---|---|---|---|---|
| $n =$ | $10$ | $10^2$ | $10^3$ | $10^4$ | $10^5$ |
| $T_1(n) =$ | $10^2$ | $10^4$ | $10^6$ | $10^8$ | $10^{10}$ |
| $T_2(n) =$ | $10^3$ | $10^4$ | $10^5$ | $10^6$ | $10^7$ |

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

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

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

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

$$T_2(n) = 100n$$

| | | | | | |
|---|---|---|---|---|---|
| $n =$ | $10$ | $10^2$ | $10^3$ | $10^4$ | $10^5$ |
| $T_1(n) =$ | $10^2$ | $10^4$ | $10^6$ | $10^8$ | $10^{10}$ |
| $T_2(n) =$ | $10^3$ | $10^4$ | $10^5$ | $10^6$ | $10^7$ |

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

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

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

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

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

$$T(n) = c_1 + c_2(t_2+t_3+\ldots+t_n) + c_3(n-1)$$

pesymistycznie:

$$T(n) = c_1 + c_2(1+2+\ldots+(n-1)) + c_3(n-1) =$$

$$= c_1 + c_2 n(n-1)\cdot\tfrac{1}{2} + c_3(n-1) =$$

$$= c_1 + \tfrac{1}{2}\cdot c_2 n^2 - \tfrac{1}{2}\cdot c_2 n + c_3(n-1) =$$

$$= \tfrac{1}{2}\cdot c_2 n^2 + (c_3 - \tfrac{1}{2}\cdot c_2)\cdot n + c_1 - c_3 =$$

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

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: $T_p(n) = \Omega(f(n))$, lub 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: $T_o(n) = \Omega(f(n))$, lub z obu stron: $T_o(n) = \Theta(f(n))$

# Przykłady i przydatne fakty

jeżeli $T_1(n) = O(f(n))$ i $T_2(n) = O(f(n))$

*to* $T_1(n) + T_2(n) = O(f(n))$

jeżeli $T(n) = O(f(n))$ i $f(n) = O(g(n))$

*to* $T(n) = O(g(n))$

jeżeli $T(n) = a_k n^k + a_{k-1} n^{k-1} + \ldots + a_0$

gdzie $a_k > 0$