Розробка та аналіз алгоритмів

2. Сортування включенням

- Метод сортування включенням
- RAM
- Аналіз алгоритму
- Асимптотичні позначення

Сортування включенням

Вхід: $A = \langle a_1, \ldots, a_n \rangle$

Вихід: $A' = \langle a'_1, \ldots, a'_n \rangle$, $a'_1 \le a'_2 \le \ldots \le a'_n$

$A = [5, 2, 4, 6, 1, 3]$

$A' = [1, 2, 3, 4, 5, 6]$

$A =$ | 5 | 2 | 4 | 6 | 1 | 3 |

| 2 | 5 | 4 | 6 | 1 | 3 |

| 2 | 4 | 5 | 6 | 1 | 3 |

| 2 | 4 | 5 | 6 | 1 | 3 |

| 1 | 2 | 4 | 5 | 6 | 3 |

$A' =$ | 1 | 2 | 3 | 4 | 5 | 6 |

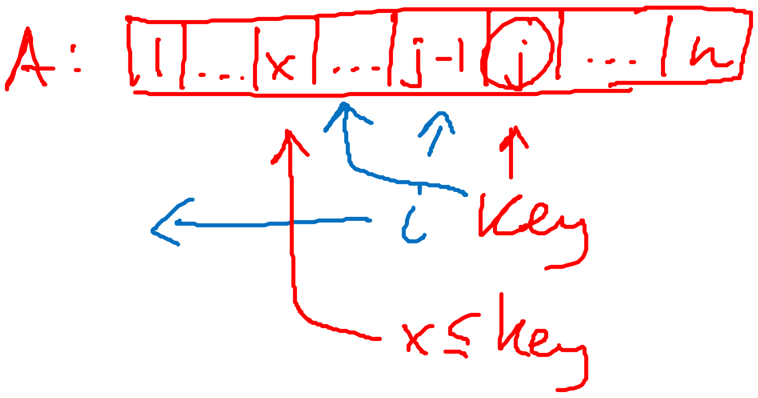
Алгоритм сортування включенням

Insertion_Sort (A)

1. for j=2 to n:
2. key = A[j]
3. i = j − 1
4. while i>0 and A[i]>key:
5. A[i+1] = A[i]
6. i = i − 1
7. A[i+1] = key

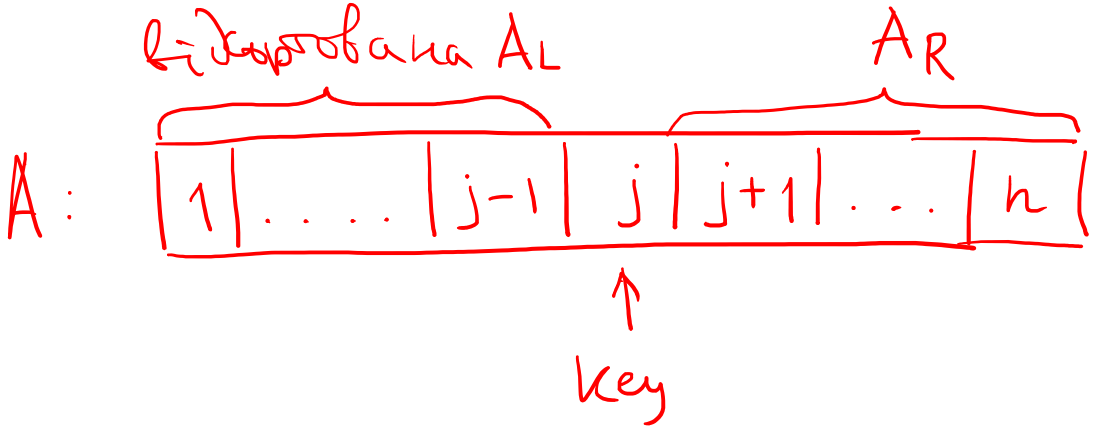
довжина масиву A — n

A[i] > key



i=0: A[i+1] = A[1] = key

Сортування включенням



$j=2$: $A_L=[a_1]$

$j=j'$: $A_L=[a'_1,\dots,a'_{j-1}]$ key $=a_j \Rightarrow A'_L=[a'_1,\dots,a'_{j-1},a'_j]$

$j=n+1$: $A_L=[a'_1,\dots,a'_n]$

Машина з довільним доступом до пам’яті (Random Access Machine, RAM)

- RAM – модель узагальненої однопроцесорної машини з довільним доступом до пам’яті
- В цій моделі команди процесору виконуються послідовно; операції, які виконуються одночасно, відсутні
- Модель містить типові команди процесору: арифметичні операції, операції переміщення даних, керуючі операції
- Для виконання кожної інструкції потрібен фіксований проміжок часу
- Модель має цілочисловий тип даних та тип чисел з плаваючою точкою

$T(n)$ – кількість операцій

Аналіз алгоритму сортування включенням

- Час роботи алгоритму для того або іншого входу вимірюється в кількості елементарних операцій, або «кроків», які необхідно виконати

Вхід: X, $|X| = n \Rightarrow T(n) = ?$

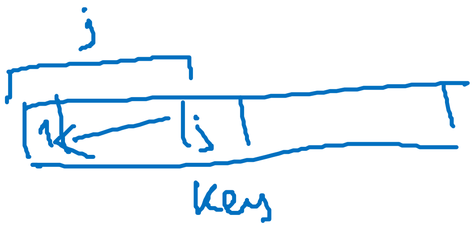
Аналіз алгоритму сортування включенням

t_j – кількість перевірок умови while, $j=2,\dots,n$

INSERTION_SORT(A)

		Час	Кількість разів
n	1. **FOR** *j* = 2 **TO** *n*:	c_1	n
$O(1)$	2. *key* = *A*[*j*]	c_2	$n-1$
$O(1)$	3. *i* = *j* − 1	c_3	$n-1$
n	4. **WHILE** *i*>0 and *A*[*i*]>*key*:	c_4	$t_2+t_3+\dots+t_n=\sum_{j=2}^{n} t_j$
$O(1)$	5. *A*[*i*+1] = *A*[*i*]	c_5	$\}\ (t_2-1)+(t_3-1)+\dots+(t_n-1)=\sum_{j=2}^{n}(t_j-1)$
$O(1)$	6. *i* = *i* − 1	c_6	
$O(1)$	7. *A*[i+1] = *key*	c_7	$n-1$
$O(n^2)$			

$$T(n)=c_1\cdot n+c_2(n-1)+c_3(n-1)+c_4\cdot\sum_{j=2}^{n}t_j+c_5\sum_{j=2}^{n}(t_j-1)+c_6\sum_{j=2}^{n}(t_j-1)+c_7(n-1)$$



$t_j=j$

Аналіз алгоритму сортування включенням

- Найкращий випадок : $t_j = 1$

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \overbrace{\sum_{j=2}^{n} t_j}^{(n-1)} + c_5 \cancelto{0}{\sum_{j=2}^{n} (t_j - 1)} + c_6 \cancelto{0}{\sum_{j=2}^{n} (t_j - 1)} + c_7(n-1) =$$

$$= n \cdot \underbrace{(c_1 + c_2 + c_3 + c_4 + c_7)}_{a} - \underbrace{(c_2 + c_3 + c_4 + c_7)}_{b} = n \cdot a - b \leftarrow \text{лінійна функція}$$

Аналіз алгоритму сортування включенням

- Найгірший випадок : $t_j = j$

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \sum_{j=2}^{n} t_j + c_5 \sum_{j=2}^{n}(t_j - 1) + c_6 \sum_{j=2}^{n}(t_j - 1) + c_7(n-1) =$$

$$\sum_{j=2}^{n} t_j = \sum_{j=2}^{n} j = \sum_{j=1}^{n} j - 1 = \frac{n(n+1)}{2} - 1 \; ; \quad \sum_{j=2}^{n}(t_j - 1) = \sum_{j=2}^{n}(j-1) = \sum_{j=1}^{n-1} j = \frac{(n-1)n}{2}$$

$$= c_1 n + c_2(n-1) + c_3(n-1) + c_4\left[\frac{n(n+1)}{2} - 1\right] + c_5 \frac{n(n-1)}{2} + c_6 \frac{n(n-1)}{2} + c_7(n-1) =$$

$$= n^2 \underbrace{\left(\frac{c_4}{2} + \frac{c_5}{2} + \frac{c_6}{2}\right)}_{a} + n \underbrace{\left(c_1 + c_2 + c_3 + \frac{c_4}{2} - \frac{c_5}{2} - \frac{c_6}{2} + c_7\right)}_{b} - \underbrace{(c_2 + c_3 + c_4 + c_7)}_{d} =$$

$$= an^2 + bn + d \rightarrow$$ квадратична функція

Аналіз алгоритму сортування включенням

- Найкращий випадок:

$A = [a_1, \ldots, a_n], \ a_1 \le a_2 \le \ldots \le a_n \Rightarrow T(n) = an + b$

- Найгірший випадок:

$A = [a_1, \ldots, a_n], \ a_1 > a_2 > \ldots > a_n \Rightarrow T(n) = an^2 + bn + d$

Порядок зростання

- В аналізі алгоритмів зазвичай досліджують час роботи алгоритму тільки у найгіршому випадку – максимальний час роботи серед усіх вхідних даних розміром *n*
  - Час роботи алгоритму в найгіршому випадку – це верхня межа для будь-яких вхідних даних
  - В деяких задачах найгірший випадок зустрічається досить часто (наприклад, пошук в базі даних елементу, якого там не існує)

$T(n) = \cancel{a}n^2 + \cancel{b}n + \cancel{d}$ $\quad n \to \infty$ $\quad n = 10^6 : T(n) = a \cdot 10^{12} + b \cdot 10^6 + d$

$a, b, d = 10^6$

$T(n) \simeq n^2, \; n \to \infty$ $\quad n = 10^9 : T(n) = 10^6 \cdot 10^{18} + 10^6 \cdot 10^9 + 10^6$

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

Асимптотичні позначення

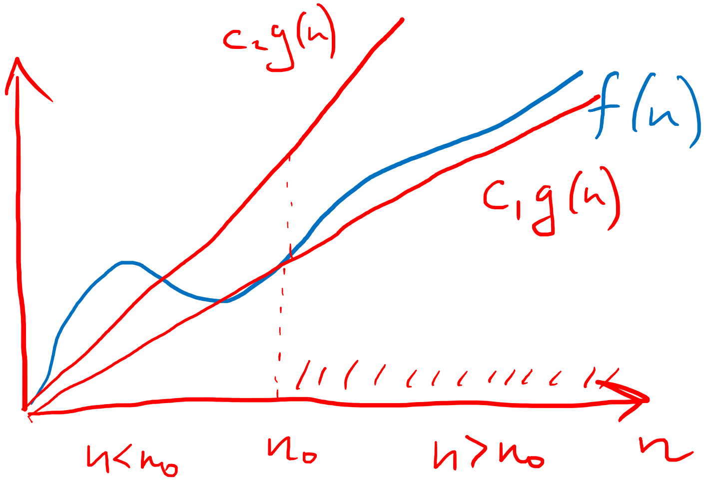
$n \to \infty$

$$\Theta(g(n)) = \{ f(n) : \exists c_1, c_2, n_0 > 0 : 0 \le c_1 g(n) \le f(n) \le c_2 g(n), \forall n \ge n_0 \}$$

$f(n) \in \Theta(g(n))$

$f(n) = \Theta(g(n))$

Множина функцій $f(n)$



$g(n)$ – асимптотично точна оцінка $f(n)$

$$f(n) = \frac{n^2}{2} - 3n \in \Theta(n^2)$$

$$c_1 n^2 \le \frac{n^2}{2} - 3n \le c_2 n^2, \quad \forall n \ge n_0$$

$$c_1 \le \frac{1}{2} - \frac{3}{n} \le c_2$$

$c_1 = \frac{1}{14}, n \ge 7$

$c_2 = \frac{1}{2}, n \ge 1 \Rightarrow$

$$\boxed{\begin{aligned} c_1 &= 1/14 \\ c_2 &= 1/2 \\ n_0 &= 7 \end{aligned}}$$

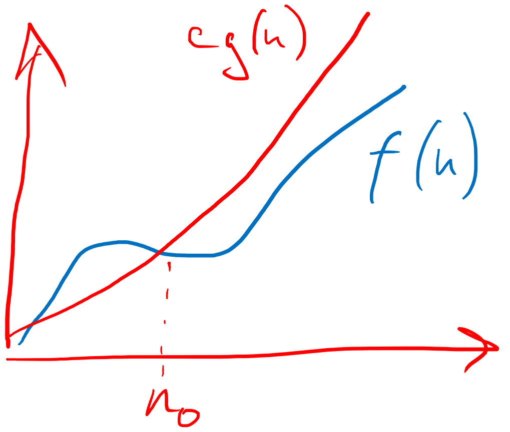
$$\boxed{\frac{1}{14} n^2 \le \frac{n^2}{2} - 3n \le \frac{1}{2} n^2, \quad n \ge 7}$$

Асимптотичні позначення

Верхня межа:

$O(g(n)) = \{f(n) : \exists c, n_0 > 0 : 0 \le f(n) \le c \cdot g(n), \forall n \ge n_0\}$

Якщо $f(n) = \Theta(g(n))$, то $f(n) = O(g(n))$

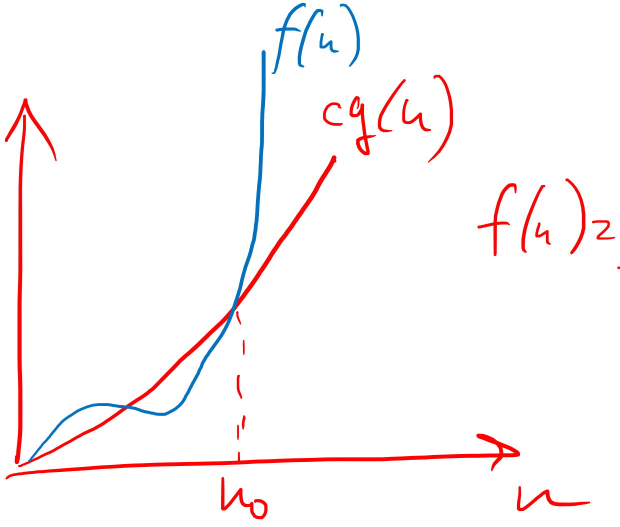


$f(n) = O(g(n))$

Нижня межа:

$\Omega(g(n)) = \{f(n) : \exists c, n_0 > 0 : cg(n) \le f(n), \forall n \ge n_0\}$

Якщо $f(n) = \Theta(g(n))$, то $f(n) = \Omega(g(n))$



$f(n) = \Omega(g(n))$