CORRETTEZZA & PERFORMANCE ⟶ RISORSE: Runtime (Tempo Computazionale)
memoria
larghezza di banda

Modello: Random-Access Machine (RAM)

Il modello RAM non tiene conto delle gerarchie di memoria

IMPORTANTE:
A noi interessa il come cresce il tempo computazionale in funzione della lunghezza dell'input

la misura dell'input dipende alla struttura dati utilizzata e dai dati utilizzati

Tempo computazionale di un algoritmo (Runtime)

Numero di istruzioni elementari e di accessi ai dati eseguiti, in funzione della lunghezza dell'input

INSERTIONSORT		Costo
1. for i=2 to n	1. c_1	n
2. key = A[i]	2. c_2	$n-1$
3. j = i-1	3. c_3	$n-1$
4. while j>0 and A[j] > key	4. c_4	$\sum_{i=2}^{n} t_i$
5. A[j+1] = A[j]	5. c_5	$\sum_{i=2}^{n} (t_i - 1)$
6. j = j+1	6. c_6	$\sum_{i=2}^{n} (t_i - 1)$
7. A[j+1]	7. c_7	$n-1$

- t_i = tempo che impiega il ciclo while, $\forall i \in \{2, \dots, n\}$

$T(n) = c_1 n + c_2 (n-1) + c_3(n-1) + c_4 \sum t_i + c_5 \sum (t_i-1) + c_6 \sum (t_i-1) + c_7 (n-1)$

Best case: (array gia ordinato)
$t_i = 1 \quad \forall i \in \{2, \dots, n\}$
$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4(n-1) + c_7(n-1) =$
$= \underbrace{(c_1+c_2+c_3+c_4+c_7)}_{a} n + \underbrace{(-c_2-c_3-c_4-c_7)}_{b}$
= a + b ---> tempo lineare

Worst case: (array gia ordinato ma in maniera inversa)
$t_i = i \quad i \in \{2, \dots, n\}$
$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \sum_{i=2}^{n} i + c_5 \sum_{i=2}^{n} (i-1) + c_6 \sum_{i=2}^{n} (i-1) + c_7(n-1) =$

$\sum_{i=2}^{n} i = (n(n+1)) / 2$

Ci interessa: l'ordine di grandezza (o tasso di crescita)
(non ci dobbiamo preoccupare troppo delle costanti)

Un algoritmo è più efficiente di un altro se il tempo computazionale nel caso peggiore (Worst case) cresce più lentamente

La notazione Big-Oh

O -> denota un upper bound asintotico

$f(n) = 7n^3 + 10n^2 - 20n + 6$ ---> non cresce piu velocemente di $8n^3$

$f(n) = O(n^3)$, $f(n) = O(n^4)$, $O(n^5)$, $O(n^3 \log n)$

$$O(g(n)) = \left\{ f(n) \;\middle|\; \begin{array}{l} Ec \in Q>o, \\ En_0 \in N : \\ \forall n >= n_0 \\ 0 <= f(n) <= g(n) \end{array} \right\}$$

$\underline{\Omega}$-> denota un lower bound asintotico

cresce almeno come $7n^3$ (asintoticamente)

$F(n) = \underline{\Omega}(n^3)$, $f(n) = \underline{\Omega}(n^2)$, $f(n) = \underline{\Omega}(n)$

$$\Omega(n) = \left\{ f(n) \;\middle|\; \begin{array}{l} Ec \in Q>o, \\ En_0 \in N : \\ \forall n >= n_0 \\ 0 <= g(n) <= f(n) \end{array} \right\}$$

$f(n) = 7n^3 + 10n^2 - 20n + 6$

Θ : denota un limite stretto

$$\Theta(g(n)) = \left\{ f(n) \;\middle|\; \begin{array}{l} Ec1, c2 \in Q>o, \\ En_0 \in N : \\ \forall n >= n_0 \\ 0 <= c1\, g(n) <= f(n) <= c2 g(n) \end{array} \right\}$$

f(n) cresce: non piu velocemente di $8n^3$

e non piu lentamente di $7n^3$

$$f(n) = \Theta(g(n)) \quad <==> \begin{cases} f(n) = 0(g(n)) \\ f(n) = \Omega(g(n)) \end{cases}$$

Little-Oh

$2n =$	$O(n^2)$	$2n =$	$O(n^2)$
$2n^2 =$	$O(n^2)$	$2n^2$!=	$O(n^2)$

O -> denota un upper bound non asintoticamente precisi (stretti)

$$O(g(n)) = \left\{ f(n) \;\middle|\; \begin{array}{l} Ec1, c2 \in Q>o, \\ En_0 \in N : \\ \forall n >= n_0 \\ 0 <= f(n) <= cg(n) \end{array} \right\}$$

ω -> denota un lower bound non asintoticamente precisi (non stretti)

O

$2n =$	$\omega(n)$	$2n$!=	$\omega(n)$
$2n^2 =$	$\omega(n)$	$2n^2 =$	$\omega(n)$

$$\omega(g(n)) = \left\{ f(n) \;\middle|\; \begin{array}{l} Ec1, c2 \in Q>o, \\ En_0 \in N : \\ \forall n >= n_0 \\ 0 <= c\, g(n) < f(n) \end{array} \right\}$$