

Selection sort

venerdì 18 marzo 2022 14:19

Qui noi facciamo i pazzi alla ricerca di ogni volta il valore minimo.
Prima cerchiamo il minimo dalla 1 posizione fino ad N e, quando trovato
Lo mettiamo come primo. Poi da 2 fino a N, fino alla fine.

Es. V = [5,3,1]

5	3	1	->	1	5	3
---	---	---	----	---	---	---

->	1	3	5
----	---	---	---

Algoritmo

Void selSort(V[])

For i = 1 to n - 1:

Pmin = i

For j = i + 1 to n:

If v[j] < v[Pmin]:

Pmin = j

App = V[i]

V[i] = V[Pmin]

V[Pmin] = App

C1(n-1)

C2(n-1)

C3(n-i) -> $C3 * \sum_{i=1}^{n-1} i$

C4*somma

C5*Tif

C6*(n-1)

C7*(n-1)

C8*(n-1)

$$T_{selSort}(n) = (c_1 + c_2 + c_6 + c_7 + c_8)(n-1) + (c_3 + c_4) * \sum_{i=1}^{n-1} i + c_5 * T_{if} \quad \xrightarrow{n-1} \frac{(n-1) * n}{2}$$

$$= (5C)(n-1) + \frac{2c(n-1)}{2} + C_5 * T_{if}$$

Caso migliore:

L'array è già ordinato = if mai testato

$t_{if} = 0$

$$T_{migliore}(n) = 5C(n-1) + 2c * \frac{n^2}{2} - 2c * \frac{n}{2} + 0 \sim 2c * \frac{n^2}{2} \sim 2cn^2 \sim cn^2 \sim n^2$$

Caso peggiore:

L'if è sempre vero = l'array è contrariamente ordinato

$$t_{if} = \sum_{i=1}^n i$$

$$T_{peggiore}(n) = 5c(n-1) + 3c * \frac{n^2}{2} - 3c * \frac{n}{2} \sim n^2$$