

SortinoSort

Il professor Sortino ha inventato un nuovo algoritmo di ordinamento.

- Il vettore di input viene diviso in tre parti, di dimensioni circa $n/3$.
- Vengono ordinati ricorsivamente i primi due terzi, i secondi due terzi, e infine di nuovo i primi due terzi.

```
SortinoSort(int[] A, int i, int j)
```

```
if  $j - i + 1 \leq 6$  then
```

```
    InsertionSort(A, i, j)
```

```
else
```

```
    int  $s = \lceil (j - i + 1) / 3 \rceil$ 
```

```
    SortinoSort(A, i, i + 2s - 1)
```

```
    SortinoSort(A, i + s, j)
```

```
    SortinoSort(A, i, i + 2s - 1)
```

SortinoSort

- ① Qual è la complessità di questo algoritmo? Il Prof. Sortino finirà nella prossima edizione del mio libro?
- ② (Difficile, Opzionale) Dimostrare per induzione che questo algoritmo è corretto. Per comodità, assumete pure che tutti i valori siano distinti.

Spoiler alert!

La complessità è rappresentata dalla seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 3T(\frac{2}{3}n) + 1 & n > 6 \\ 1 & n \leq 6 \end{cases}$$

Utilizzando il teorema delle ricorrenze lineari con partizione bilanciata, si ottiene che $a = 3$, $b = 3/2$, da cui $\alpha = \log_{3/2} 3$; inoltre, $\beta = 0$. Siamo quindi nel caso $T(n) = n^\alpha$.

Non avete una calcolatrice e non sapete quanto sia $\log_{3/2} 3$?

La complessità è rappresentata dalla seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 3T(\frac{2}{3}n) + 1 & n > 6 \\ 1 & n \leq 6 \end{cases}$$

Utilizzando il teorema delle ricorrenze lineari con partizione bilanciata, si ottiene che $a = 3$, $b = 3/2$, da cui $\alpha = \log_{3/2} 3$; inoltre, $\beta = 0$. Siamo quindi nel caso $T(n) = n^\alpha$.

Non avete una calcolatrice e non sapete quanto sia $\log_{3/2} 3$?

E' semplice: $(\frac{3}{2})^2 = \frac{9}{4} < 3$, mentre $(\frac{3}{2})^3 = \frac{27}{8} > 3$. Quindi α è compreso fra 2 e 3, e quindi questo algoritmo è addirittura peggiore di Insertion Sort. Il prof. Sortino non finirà nel mio libro.