

Statistiche d'ordine

Come faccio a trovare il numero più piccolo in un array disordinato? Guardali tutti

$T_{min} \rightarrow \theta(n)$ Quante domande mi faccio? n

Come faccio a trovare il numero più grande in un array disordinato? Guardali tutti

$T_{max} \rightarrow \theta(n)$ Quante domande mi faccio? n

E cercarli contemporaneamente?

$T_{minmax} \rightarrow \theta(n)$ Quante domande mi faccio? $2n$

Come faccio a velocizzare?

Confronto una coppia di num alla volta \rightarrow confronto il num più piccolo e il più grande
Tempo $\rightarrow \frac{3}{2}n$ con il minimo con il massimo

Definizione

- minimo : prima statistica d'ordine
- massimo : n-esima statistica d'ordine
- i-esima statistica d'ordine : i-esimo numero più piccolo

Oss

come faccio a trovare l'n-esima statistica d'ordine?

- 1) Ordino l'array e considero l'n-esimo elemento
- 2) Calcolo n volte il minimo e tolgo all'array l'elemento minimo trovato
- 3) Con un pivot divido l'array tra più grandi e più piccoli, in base a quanti elementi ci sono nelle due partizioni scelgo quale considerare e scarto l'altra.

Pseudocodice :

```
int StatOrd(A[], I, F, i-esima) {  
    If I < F {  
        Q = Partition(A[], I, F)  $\rightarrow$  e' importante  
        If i-esima <= Q {  
            R = StatOrd(A[], I, Q, i-esima)  
        } else {  
            R = StatOrd(A[], Q+1, F, i-esima - Q)  
        }  
        Return(R)  
    } else {  
        // ...  
    }
```

```

}
}
Return (I)      caso base

```

Calcolo dei tempi

$$T(n) = T\left(\frac{n}{2}\right) + \theta(n)$$

metodo dell'esperto

$$\longrightarrow a = 1 \quad b = 2$$
$$n^{\log_b a} = n^{\log_2 1} = n^0 = 1$$

$\Theta(n) \leadsto$ mediamente

il caso peggiore è molto raro