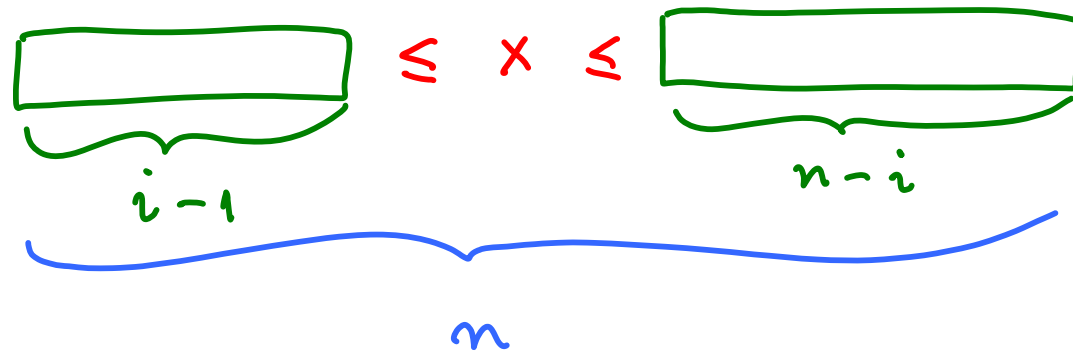


MEDIANE E
STATISTICHE
D'ORDINE

- L' i -ESIMA STATISTICA D'ORDINE DI UN INSIEME DI n ELEMENTI
E' L' i -ESIMO ELEMENTO PIÙ PICCOLO,
CIOE' UN ELEMENTO x DELL'INSIEME TALE CHE I RIMANENTI
($n-i$) ELEMENTI POSSONO ESSERE COSÌ SUDDIVISI:



CASI PARTICOLARI

- $i=1$ MINIMO

- $i=n$ MASSIMO

- $i = \left\lfloor \frac{n+1}{2} \right\rfloor$ MEDIANA (INFERIORE)

- $i = \left\lceil \frac{n+1}{2} \right\rceil$ MEDIANA SUPERIORE

PROBLEMA DELLA SELEZIONE

INPUT: - UN INSIEME A DI n NUMERI (DISTINTI)
- UN INTERO $1 \leq i \leq n$

OUTPUT: L' i -ESIMA STATISTICA D'ORDINE DI A

ALGORITMO BANALE:

- SI ORDINI A
- SI RESTITUISCA L'ELEMENTO $A[i]$

COMPLESSITA': $O(n \log n)$

- PRESENTEREMO UN ALGORITMO LINEARE

ALCUNI CASI PARTICOLARI

MINIMUM(A)

```
1  min = A[1]
2  for i = 2 to A.length
3      if min > A[i]
4          min = A[i]
5  return min
```

CONFRONTI ESEGUITI = $n-1$

ESERCIZIO DIMOSTRARE CHE SONO NECESSARI $(n-1)$ CONFRONTI

- ANALOGAMENTE PER IL MASSIMO

CALCOLARE MINIMO E MASSIMO DI n ELEMENTI

I SOLUZIONE

- CALCOLARE IL MINIMO DI n ELEMENTI
- CALCOLARE IL MASSIMO DI $(n-1)$ ELEMENTI

$$\# \text{ CONFRONTI} = (n-1) + (n-2) = 2n-3$$

- C'E' UNA SOLUZIONE PIÙ EFFICIENTE (IN TERMINI DI NUMERO DI CONFRONTI) ?

II SOLUZIONE

- SIA A UN INSIEME DI n NUMERI,
- SI SUDDIVIDANO GLI n ELEMENTI IN $\frac{n}{2}$ COPPIE (PIÙ UN EVENTUALE ELEMENTO SPAIATO)
- SI CONFRONTINO GLI ELEMENTI DI CIASCUNA COPPIA, METTENDO I VINCENTI IN UN INSIEME B E I PERDENTI IN UN INSIEME C
- SE C'E' UN ELEMENTO SPAIATO, LO SI AGGIUNGA SIA A B CHE A C .
- CHIARAMENTE
$$\max B = \max A$$
$$\min C = \min A$$
- SI DETERMININO QUINDI $\max B$ E $\min C$

CONFRONTI :

- SE n E' PARI :

$$\frac{n}{2} + \underbrace{\left(\frac{n}{2} - 1\right)}_{\text{max B}} + \underbrace{\left(\frac{n}{2} - 1\right)}_{\text{min C}} = \frac{3n}{2} - 2 = \left\lceil \frac{3n}{2} \right\rceil - 2$$

- SE n E' DISPARI :

$$\frac{n-1}{2} + \left[\left(\frac{n-1}{2} + 1 \right) - 1 \right] + \left[\left(\frac{n-1}{2} + 1 \right) - 1 \right] = \left(\frac{3n}{2} + \frac{1}{2} \right) - 2$$
$$= \left\lceil \frac{3n}{2} \right\rceil - 2$$

DUNQUE, IN DEFINITIVA :

$$\# \text{ CONFRONTI} = \left\lceil \frac{3n}{2} \right\rceil - 2$$

NOTA: SI DIMOSTRA CHE $\left\lceil \frac{3n}{2} \right\rceil - 2$ CONFRONTI SONO NECESSARI

SELEZIONE IN TEMPO ATTESO LINEARE

RANDOMIZED-SELECT(A, p, r, i)

```
1  if  $p == r$ 
2      return  $A[p]$ 
3   $q =$  RANDOMIZED-PARTITION( $A, p, r$ )
4   $k = q - p + 1$ 
5  if  $i == k$            // the pivot value is the answer
6      return  $A[q]$ 
7  elseif  $i < k$ 
8      return RANDOMIZED-SELECT( $A, p, q - 1, i$ )
9  else return RANDOMIZED-SELECT( $A, q + 1, r, i - k$ )
```

- SI DIMOSTRA CHE IL TEMPO ATTESO DI RANDOMIZED-SELECT

È $O(n)$

(CASO PESSIMO: $O(n^2)$)

SELEZIONE IN TEMPO LINEARE (NEL CASO PECCIORE)

SELECT(A, i)

- SI DIVIDA A IN $\lceil \frac{n}{5} \rceil$ GRUPPI DI 5 ELEMENTI, PIÙ UN
EVENTUALE GRUPPO CON I RESTANTI $n \bmod 5$ ELEMENTI

- SI DETERMININO LE MEDIANE DEGLI $\lceil \frac{n}{5} \rceil$ GRUPPI E SIA

M LA SEQUENZA DI TALI MEDIANE

- SI EFFETTUÌ LA CHIAMATA RICORSIVA $m = \text{SELECT}(M, \lceil \frac{|M|}{2} \rceil)$

- SI PARTIZIONI A IN A_1, A_2, A_3 DOVE

$$A_1 = [x \in A : x < m]$$

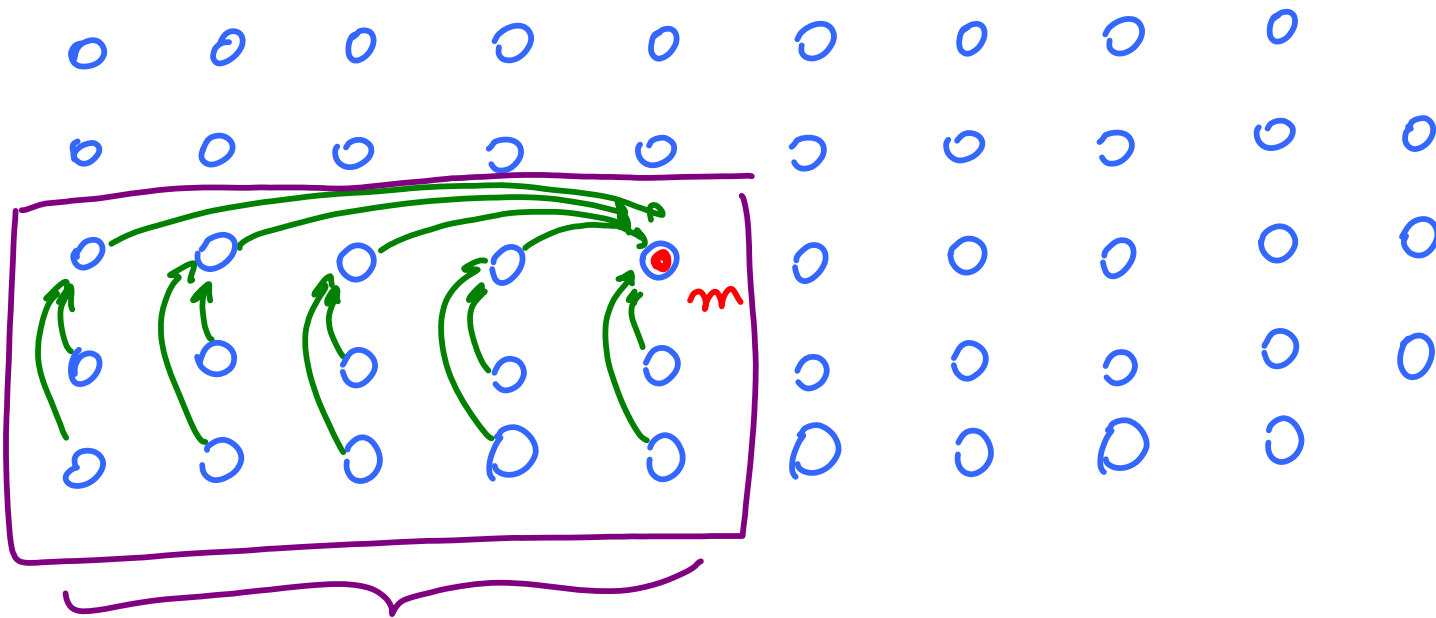
$$A_2 = [x \in A : x = m]$$

$$A_3 = [x \in A : x > m]$$

- if $|A_1| \geq i$
then return $\text{SELECT}(A_1, i)$
else if $(|A_1| + |A_2| \geq i)$
then return m
else return $\text{SELECT}(A_3, i - |A_1| - |A_2|)$

CORRETTEZZA: PGR INDUZIONE SU $|A|$

ANALISI DI COMPLESSITA'

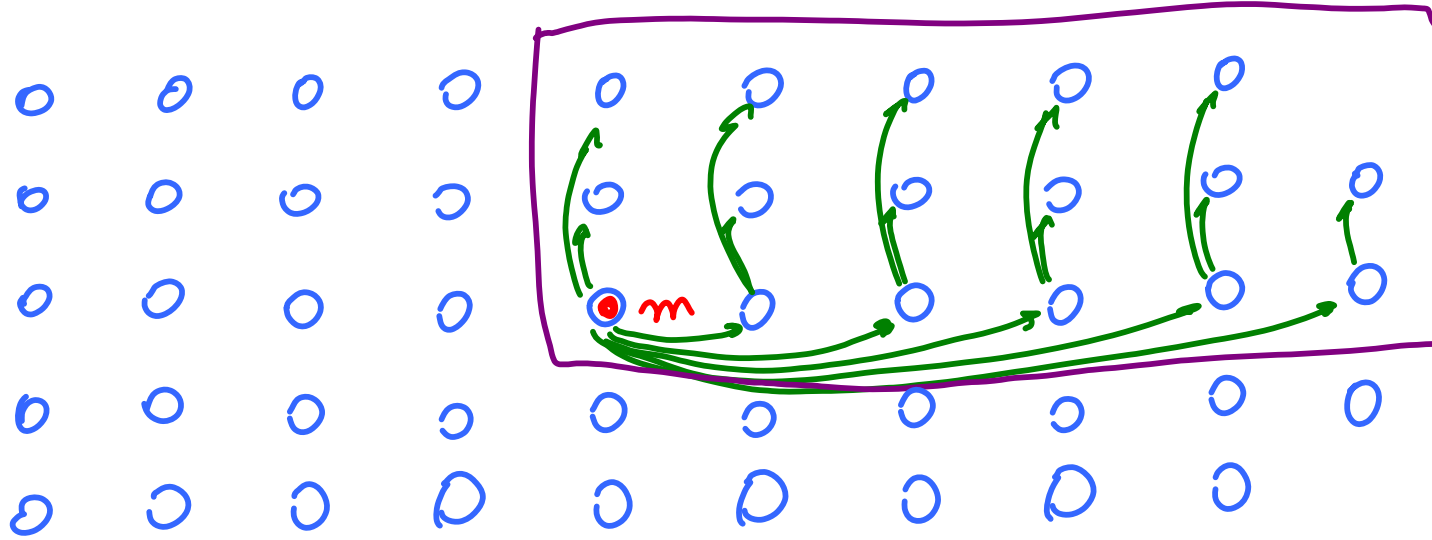


$$\lceil \frac{m}{5} \rceil / 2$$

- LIMITE SUPERIORE SUL NUMERO DI ELEMENTI $> m$

$$\text{SI HA: } |A_1| + |A_2| \geq 3 \cdot \lceil \frac{m}{5} \rceil / 2 - 2 = 3 \lceil \frac{m}{10} \rceil - 2 \geq \frac{3n}{10} - 2$$

$$\text{QUINDI: } |A_3| = n - (|A_1| + |A_2|) \leq n - \frac{3n}{10} + 2 = \frac{7n}{10} + 2$$



$$\left\lfloor \left\lfloor \frac{m}{5} \right\rfloor / 2 \right\rfloor + 1 \leq \frac{7m}{10} + 2$$

- LIMITE SUPERIORE SUL NUMERO DI ELEMENTI $< m$

$$|A_2| + |A_3| \geq 3 \cdot \left(\left\lfloor \left\lfloor \frac{m}{5} \right\rfloor / 2 \right\rfloor + 1 \right) - 2 \geq 3 \cdot \left\lfloor \left\lfloor \frac{n}{5} \right\rfloor / 2 \right\rfloor - 2 = 3 \left\lfloor \frac{n}{10} \right\rfloor - 2$$

$$\geq \frac{3n}{10} - 2$$

- QUINDI : $|A_1| = n - (|A_2| + |A_3|) \leq n - \frac{3n}{10} + 2 = \frac{7n}{10} + 2$

- PERTANTO IL TEMPO DI ESECUZIONE $T(n)$ DELL'ALGORITMO SELECT SODDISFA LA SEGUENTE RICORRENZA:

$$T(n) \leq T\left(\left\lfloor \frac{n}{5} \right\rfloor\right) + T\left(\frac{7n}{10} + 2\right) + \Theta(n).$$

- SOLUZIONE CON IL METODO DI AKRA-BAZZI
- VERIFICA PER INDUZIONE

- APPLICHIAMO INIZIALMENTE IL METODO DI AKRA-BAZZI PER RISOLVERE

$$T(n) \leq T\left(\left\lfloor \frac{n}{5} \right\rfloor\right) + T\left(\frac{7n}{10} + 2\right) + \Theta(n)$$

- SI HA: $a_1 = a_2 = 1 > 0$

$$b_1 = \frac{1}{5}, \quad b_2 = \frac{7}{10}$$

$$h_1(n) = 0, \quad h_2(n) = 2$$

$$g(n) = \Theta(n) = \Theta(n^1), \quad c = 1$$

- SIA p TALE CHE $\left(\frac{1}{5}\right)^p + \left(\frac{7}{10}\right)^p = 1$.

OCCORRE CONFRONTARE p CON c .

$$\left(\frac{1}{5}\right)^1 + \left(\frac{7}{10}\right)^1 = \frac{1}{5} + \frac{7}{10} = \frac{2+7}{10} = \frac{9}{10} < 1, \quad \text{E POICHE' LA}$$

FUNZIONE $\left(\frac{1}{5}\right)^x + \left(\frac{7}{10}\right)^x$ E' DECRESCENTE, NE SEGUE CHE

$p < 1 = c$. PERTANTO $T(n) = \Theta(n)$.

- DIMOSTRIAMO PER INDUZIONE CHE

$$T(n) \leq cn, \text{ PER } n \text{ SUFF. GRANDE} \\ \text{E PER QUALCHE } c > 0$$

$$T(n) \leq T\left(\left\lceil \frac{n}{5} \right\rceil\right) + T\left(\frac{7n}{10} + 2\right) + \Theta(n)$$

$$\leq c \left\lceil \frac{n}{5} \right\rceil + c \left(\frac{7n}{10} + 2\right) + an$$

(PER QUALCHE $a > 0$)
(n SUFF. GRANDE)

$$< c \frac{n}{5} + c + \frac{7cn}{10} + 2c + an = \frac{9cn}{10} + 3c + an$$

$$= cn + \left(-\frac{cn}{10} + 3c + an\right)$$

- IN PARTICOLARE, SE $c > 40a$,

$$\frac{c}{10} > 4a \implies -\frac{c}{10} < -4a \implies -\frac{c}{10} + a < -3a < 0$$

- PER $n \geq n_0$ (PER QUALCHE $n_0 \geq 40$)

SI HA:

$$-\frac{cn}{10} + 3c + an = \left(-\frac{c}{10} + a\right)n + 3c \leq \left(-\frac{c}{10} + a\right)40 + 3c$$

$$= -\frac{c}{10} \cdot 40 + 40a + 3c = -4c + 40a + 3c = -c + 40a < 0$$

E QUINDI

$$T(n) \leq cn, \quad \forall n \in \mathbb{N}^+$$

CON $c > \max\left(40a, T(1), \frac{T(2)}{2}, \dots, \frac{T(n_0-1)}{n_0-1}\right)$.

ESERCIZI

9.3-3

Show how quicksort can be made to run in $O(n \lg n)$ time in the worst case, assuming that all elements are distinct.

9.3-5

Suppose that you have a “black-box” worst-case linear-time median subroutine. Give a simple, linear-time algorithm that solves the selection problem for an arbitrary order statistic.

9.3-6

The k th *quantiles* of an n -element set are the $k - 1$ order statistics that divide the sorted set into k equal-sized sets (to within 1). Give an $O(n \lg k)$ -time algorithm to list the k th quantiles of a set.

9.3-7

Describe an $O(n)$ -time algorithm that, given a set S of n distinct numbers and a positive integer $k \leq n$, determines the k numbers in S that are closest to the median of S .

9.3-8

Let $X[1..n]$ and $Y[1..n]$ be two arrays, each containing n numbers already in sorted order. Give an $O(\lg n)$ -time algorithm to find the median of all $2n$ elements in arrays X and Y .

9.3-9

Professor Olay is consulting for an oil company, which is planning a large pipeline running east to west through an oil field of n wells. The company wants to connect a spur pipeline from each well directly to the main pipeline along a shortest route (either north or south), as shown in Figure 9.2. Given the x - and y -coordinates of the wells, how should the professor pick the optimal location of the main pipeline, which would be the one that minimizes the total length of the spurs? Show how to determine the optimal location in linear time.

9-1 Largest i numbers in sorted order

Given a set of n numbers, we wish to find the i largest in sorted order using a comparison-based algorithm. Find the algorithm that implements each of the following methods with the best asymptotic worst-case running time, and analyze the running times of the algorithms in terms of n and i .

- a.*** Sort the numbers, and list the i largest.
- b.*** Build a max-priority queue from the numbers, and call EXTRACT-MAX i times.
- c.*** Use an order-statistic algorithm to find the i th largest number, partition around that number, and sort the i largest numbers.