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 $(n-1)$ ELEMENTI POSSANO ESSERE COSÌ DISPOSTI:

$$\underbrace{\underbrace{\boxed{}}_{i-1} \le x \le \underbrace{\boxed{}}_{n-i}}_{n}$$

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

ESEMPIO

1	2	3	4	5	6	7
5	3	1	6	8	21	14

$\left\lfloor \frac{7+1}{2} \right\rfloor = \left\lceil \frac{7+1}{2} \right\rceil = 4$ UNICA MEDIANA

1	2	3	4	5	6	7	8
5	3	1	6	8	21	14	7

$\left\lfloor \frac{8+1}{2} \right\rfloor = 4$ MEDIANA INFERIORE

$\left\lceil \frac{8+1}{2} \right\rceil = 5$ 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

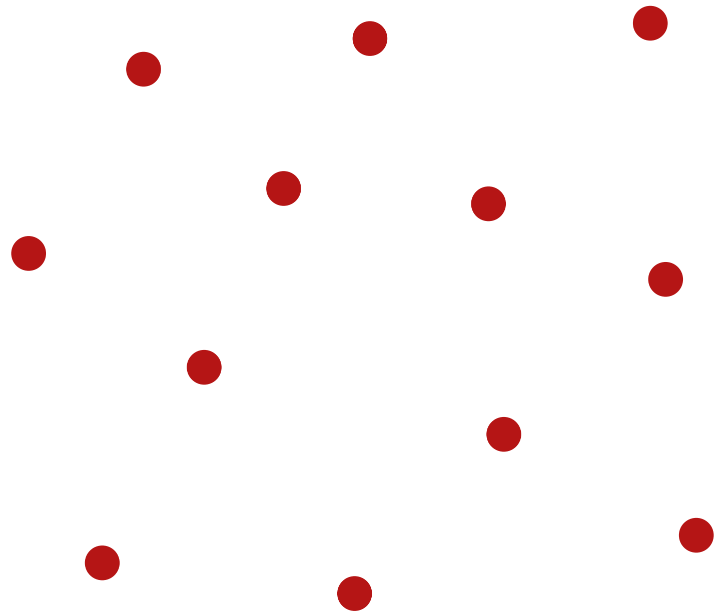
ALGORITMO BANALE:

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

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

- PRESENTEREMO UN ALGORITMO LINEARE

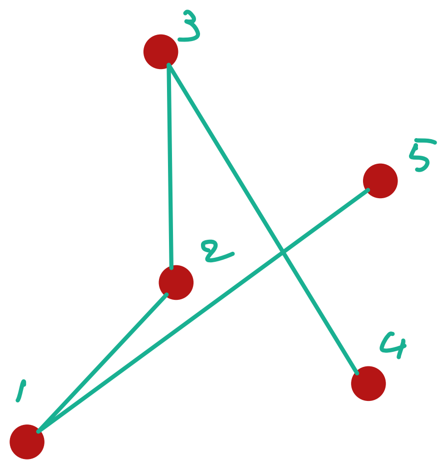
ALCUNI CASI PARTICOLARI



```
MINIMUM(A)
min = A[1]
for i = 2 to A.length
    if min > A[i]
        min = A[i]
return min
```

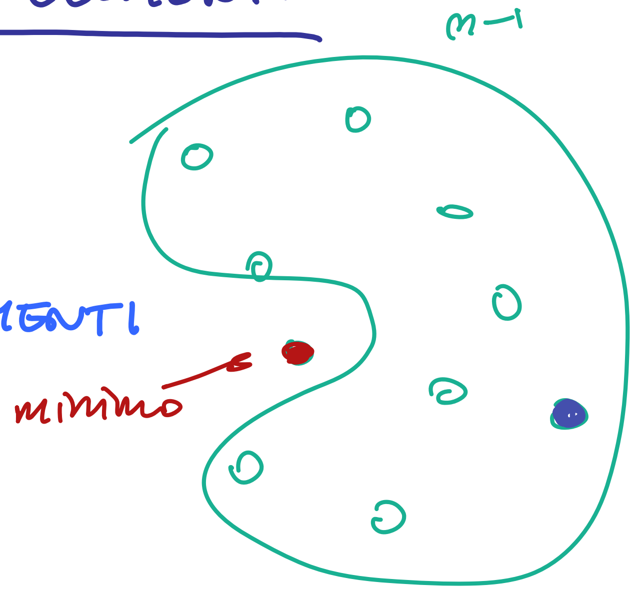
\# CONFRONTI ESEGUITI $= n-1$

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

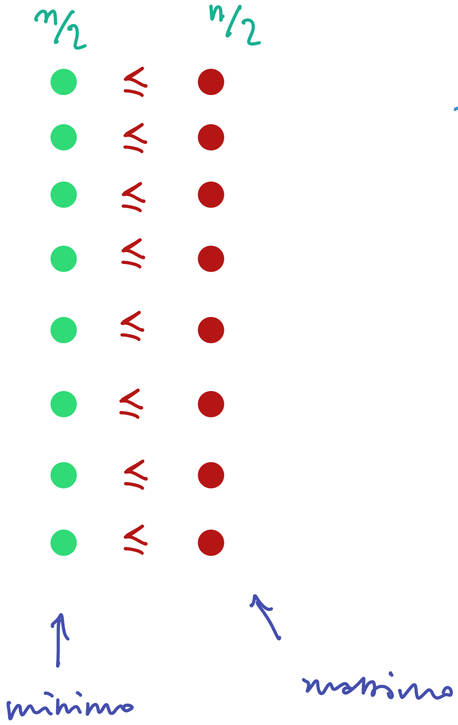
CONFRONTI = $\underline{(n-1)+(n-2) = 2n-3}$

- 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 $\min C$

n PARI

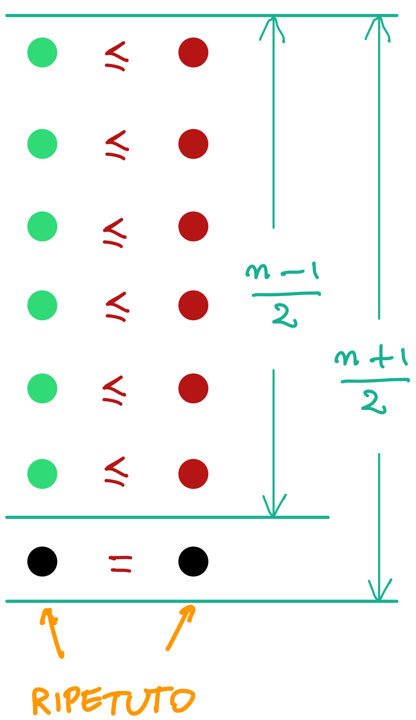


$$\#\text{CONFRONTI} = \frac{n}{2} + \left(\frac{n}{2} - 1\right) + \left(\frac{n}{2} - 1\right)$$

$$= \frac{3n}{2} - 2$$

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

n DISPARI



$$\#\text{CONFRONTI} = \frac{n-1}{2} + \left(\frac{n+1}{2} - 1\right) + \left(\frac{n+1}{2} - 1\right)$$

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

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

IN OGNI CASO

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

SELEZIONE IN TEMPO ATTESO LINEARE

```
RANDOMIZED-SELECT(A, p, r, i)
if p == r
    return A[p]
q = RANDOMIZED-PARTITION(A, p, r)
k = q - p + 1
if i == k           // the pivot value is the answer
    return A[q]
elseif i < k
    return RANDOMIZED-SELECT(A, p, q - 1, i)
else return RANDOMIZED-SELECT(A, q + 1, r, i - k)
```

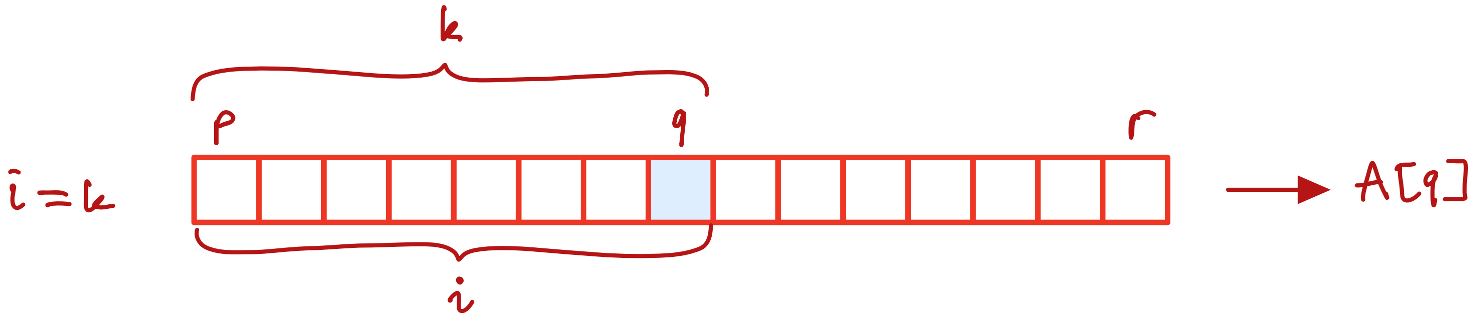
- SI DIMOSTRA CHE IL TEMPO ATTESO DI RANDOMIZED-SELECT

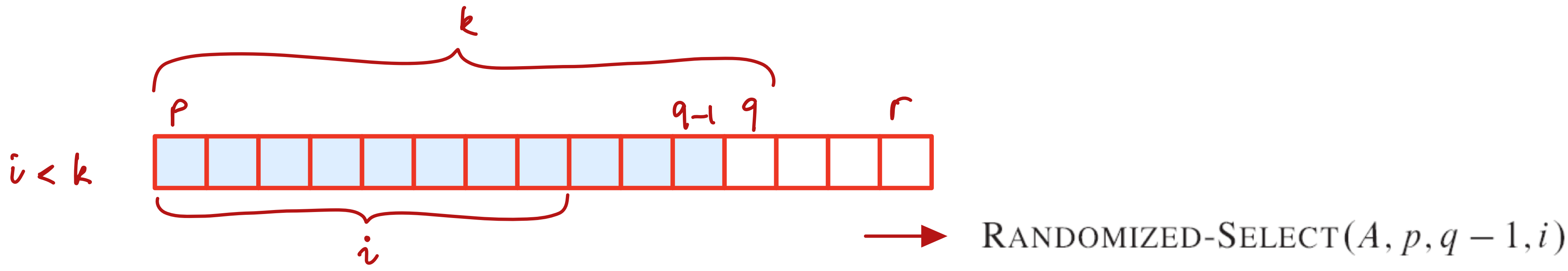
  È $O(n)$

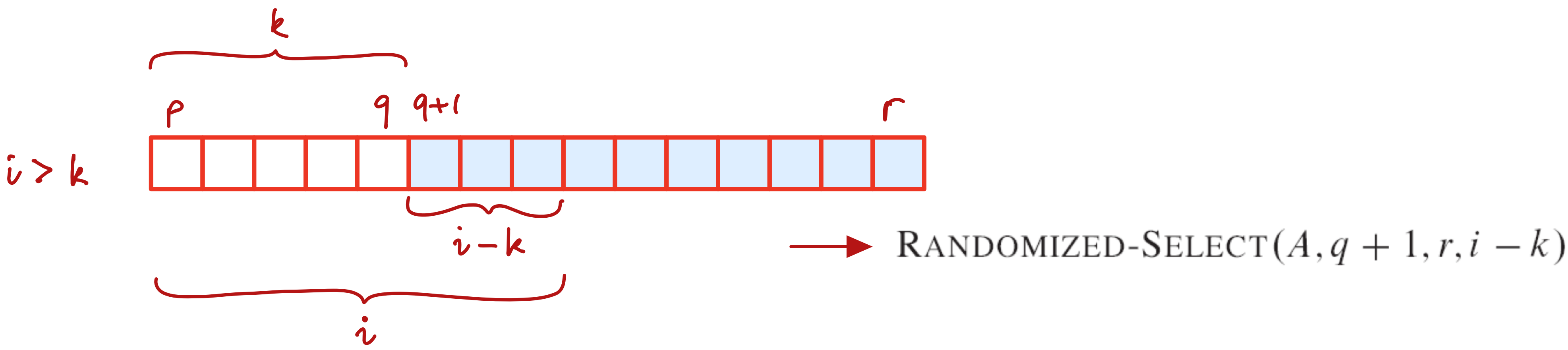
  (CASO PESSIMO: $O(n^2)$)

$q = \text{RANDOMIZED-PARTITION}(A, p, r)$

$k = q - p + 1$







SELEZIONE IN TEMPO LINEARE (NEL CASO PEGGIORE)

SELECT (A, i)

- SI DIVIDA A IN $\left\lfloor \frac{n}{5} \right\rfloor$ GRUPPI DI 5 ELEMENTI, PIÙ UN EVENTUALE GRUPPO CON I RESTANTI $n \bmod 5$ ELEMENTI $\quad O(n)$

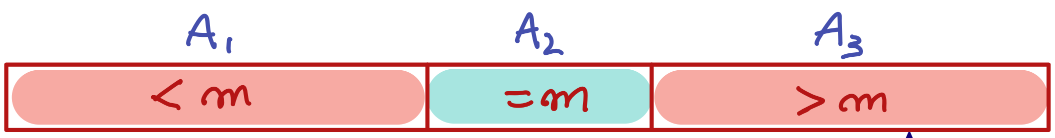
- SI DETERMININO LE MEDIANE DEGLI $\left\lceil \frac{n}{5} \right\rceil$ GRUPPI E SIA M LA SEQUENZA DI TALI MEDIANE $\quad O(n)$

- SI EFFETTUI LA CHIAMATA RICORSIVA $m = \text{SELECT}\left(M, \left\lceil \frac{|M|}{2} \right\rceil\right)$ $\quad T\left(\frac{n}{5}\right)$

- SI PARTIZIONI A IN A_1, A_2, A_3 DOVE

$$A_1 = [x \in A : x < m]$$
$$A_2 = [x \in A : x = m]$$
$$A_3 = [x \in A : x > m]$$

$O(n)$



i

- **if** $|A_1| \geq i$

  then **return** $\text{SELECT}(A_1, i)$ $\quad T\left(\frac{7n}{10}+2\right)$

  else **if** $(|A_1| + |A_2| \geq i)$ $\quad \Theta(1)$

  then **return** m

  else **return** $\text{SELECT}(A_3, i - |A_1| - |A_2|)$ $\quad T\left(\frac{7n}{10}+2\right)$

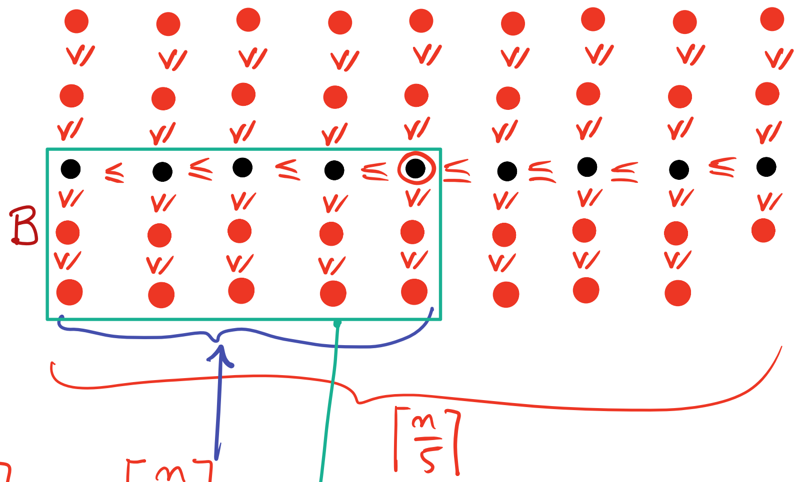
IN ALTERNATIVA

$\Downarrow$ (CASO PESSIMO)

$T\left(\frac{7n}{10}+2\right)$

CORRETTEZZA: PER INDUZIONE SU $|A|$

UN LIMITE SUPERIORE PER $|A_3|$



$$\left\lceil \frac{\lceil \frac{n}{5} \rceil}{2} \right\rceil = \left\lceil \frac{n}{10} \right\rceil$$

$$|B| \geq 3\left\lceil \frac{n}{10} \right\rceil - 2$$

POICHE' $B \subseteq A_1 \cup A_2$, SI HA

$$|A_1| + |A_2| \geq 3\left\lceil \frac{n}{10} \right\rceil - 2$$

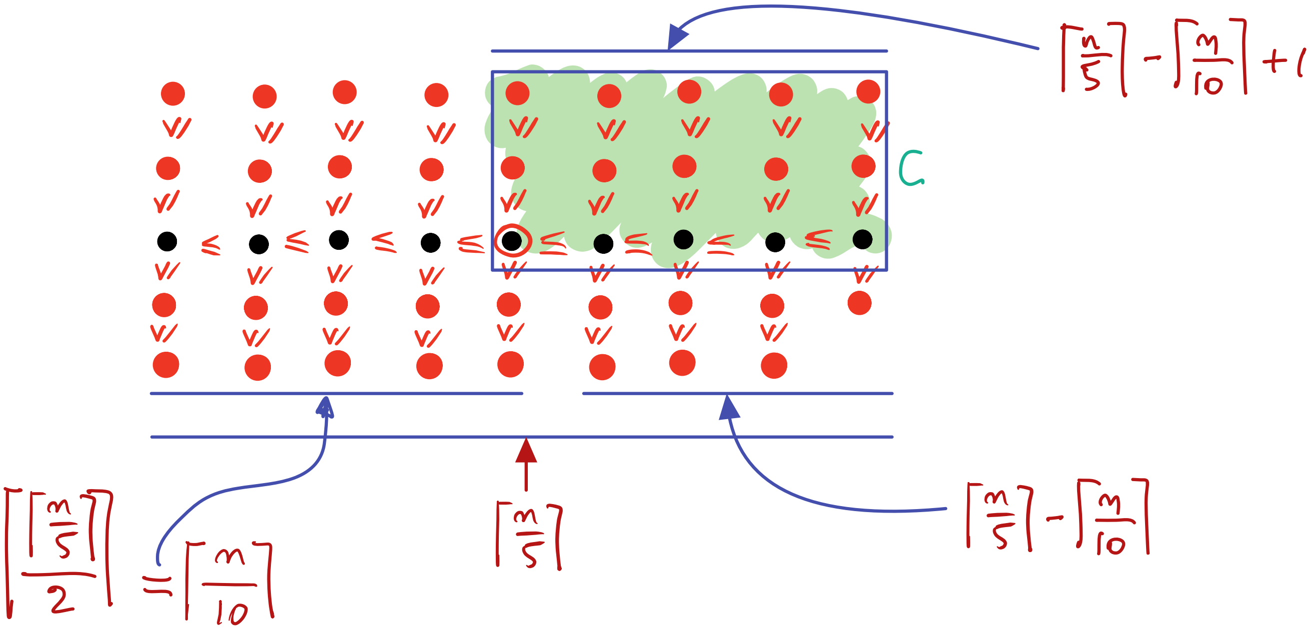
DUNQUE

$$|A_3| = n - (|A_1| + |A_2|) \leq n - 3\left\lceil \frac{n}{10} \right\rceil + 2$$

$$\leq n - \frac{3n}{10} + 2 \qquad \left(\text{IN QUANTO } -3\left\lceil \frac{n}{10} \right\rceil \leq -\frac{3n}{10} \right)$$

$$= \frac{7n}{10} + 2$$

UN LIMITE SUPERIORE PER $|A_1|$



$$|C| \geq 3\left(\left\lceil \frac{n}{5} \right\rceil - \left\lceil \frac{n}{10} \right\rceil + 1\right) - 2 > 3\left(\frac{n}{5} - \frac{n}{10}\right) - 2 = \frac{3n}{10} - 2$$

POICHE'

- $\left\lceil \frac{n}{10} \right\rceil < \frac{n}{10} + 1 \rightarrow \left\lceil \frac{n}{10} \right\rceil - 1 < \frac{n}{10} \rightarrow -\left\lceil \frac{n}{10} \right\rceil + 1 > -\frac{n}{10}$
- $\left\lceil \frac{n}{5} \right\rceil \geq \frac{n}{5}$

$$C \subseteq A_2 \cup A_3 \longrightarrow |A_2| + |A_3| > 3\frac{n}{10} - 2$$

PERTANTO $\quad -(|A_2| + |A_3|) < -\frac{3n}{10} + 2$

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

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

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

- SOLUZIONE CON IL METODO DI AKRA-BAZZI

- VERIFICA PER INDUZIONE

- APPLICHIAMO INIZIALMENTE IL METODO DI AKRA-BAZZI PER RISOLVERE

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

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

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

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

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

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

OCCORRE CONFRONTARE p CON c.

$\left(\frac{1}{5}\right)^1 + \left(\frac{7}{10}\right)^1 = \frac{1}{5} + \frac{7}{10} = \frac{2+7}{10} = \frac{9}{10} < 1$, 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

$T(n) \leq cn$, PER n SUFF. GRANDE 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)

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

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

$$= cn + \left(\left(a - \frac{c}{10}\right)n + 3c\right) \boxed{\overset{?}{\leq} cn}$$

AFFINCHE' RISULTI $T(n) \leq cn$, PER $n \geq n_0$,

E' SUFFICIENTE IMPORRE CONDIZIONI SU c ED n_0

IN MODO TALE DA ASSICURARE CHE VALGA

$$\left(a - \frac{c}{10}\right) n + 3c \leq 0, \quad \text{PER} \quad n \geq n_0.$$

IN PARTICOLARE, SE $\boxed{c > 40a}$, SI HA:

$$\frac{c}{10} > 4a \longrightarrow -\frac{c}{10} < -4a \longrightarrow a - \frac{c}{10} < -3a < 0.$$

E DUNQUE PER $n \geq 40$ SI HA:

$$\begin{aligned}\left(a - \frac{c}{10}\right) n + 3c &\leq \left(a - \frac{c}{10}\right) \cdot 40 + 3c \\ &= 40a - 4c + 3c \\ &= 40a - c \\ &< 0\end{aligned}$$

$(n_0 \geq 40)$, E QUINDI

$$T(n) = O(n).$$

MA $T(n) = \Omega(n)$.

PERTANTO: $T(n) = \Theta(n)$.

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 kth ***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 kth quantiles of a set.

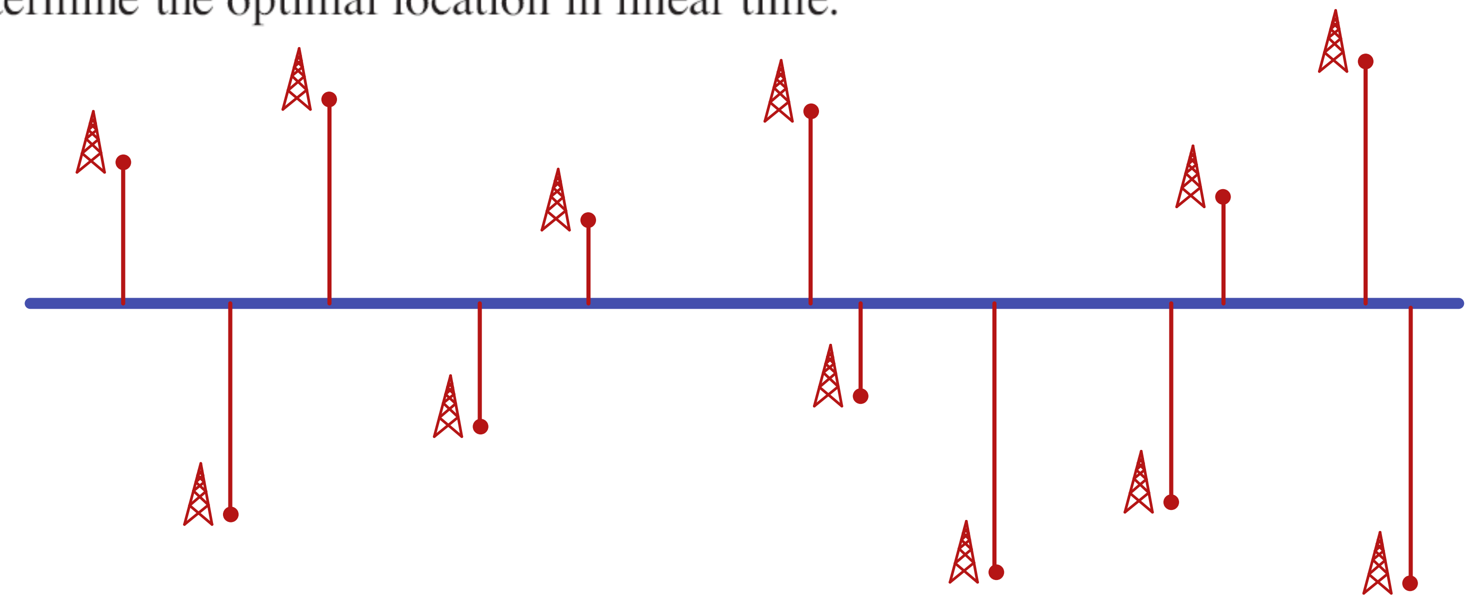
9.3-7
Describe an $O(n)$-time algorithm that, given a set S of n distinct numbers and a positive integer $k \leq n$, determines the k numbers in S that are closest to the median of S.

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 ith largest number, partition around that number, and sort the i largest numbers.