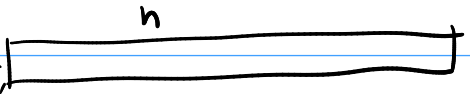
# DIVIDE et IMPERA: SELEZIONE

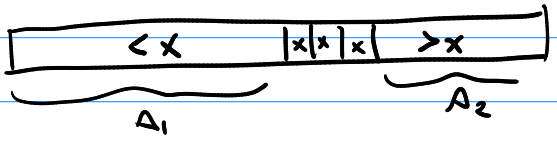


## Altri esempi

### Ripasso del Quicksort (Cap 7)

x è il PIVOT/PERNO

- dato x e A = 

- riorganizza A con 

$\Theta(n)$
- ordina $A_1$ e $A_2$ ricorsivamente

RISULTATO: A è ordinato

COMPLESSITÁ: Dipende dalla qualità della divisione di A in $A_1$ e $A_2$

$$T(n) = T(n_1) + T(n_2) + \Theta(n) \qquad |A_1| = n_1 \quad |A_2| = n$$

Caso peggiore: $|A_1| = n-1$ e $|A_2| = 1$ a tutti i livelli $T(n) = \Theta(n^2)$

Caso medio: "abbastanza spesso" nella ricorsione, SCEGLIENDO IL PIVOT A CASO abbiamo che $\frac{n}{4} \leq |A_1|, |A_2| \leq \frac{3}{4}n$.

da cui si ottiene $T(n) = \Theta(n \log n)$

# Problema della Selezione (Cap 9)

- Data una sequenza: il RANGO di $x$ elemento di $A$ è:

$$|\{a \in A,\ a \leq x\}|$$

- Se $A$ è costituito da elementi distinti
  - l'elemento di rango 1 è il MINIMO
  - l'elemento di rango $n$ è il MASSIMO
  - l'elemento di rango $\lceil n/2 \rceil$ è il MEDIANO

**PROBLEMA:** dato un array $A$ e $1 \leq k \leq n$, trovare l'elemento di rango $k$ in $A$

**Soluzione 1** ordinare $A$ e restituire $A[k-1]$

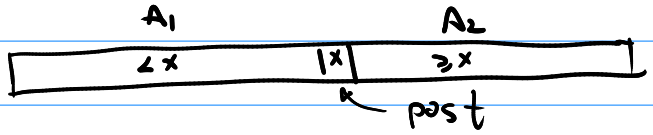
COMPLESSITÀ $\Theta(n \log n)$

**Soluzione 2** Usare una tecnica simile a quella di Quicksort (Cap 9.2)

**CASO BASE**: Se $k < 1$ o $k > len(A)$ → errore

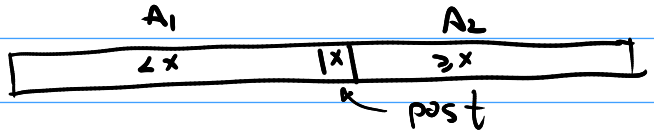
Se $len(A) = 1$ e $k = 1$ → restituisci $A[1]$

**Ricorsione** prendi $x$ in $A$ detto pivot (oppure perno) e, utilizzando la riorganizzazione di QUICKSORT, ottenere




- Se $k = t+1$ allora restituisci $x$

altrimenti...




- Se $k < t+1$, seleziona elemento di rango $k$ da $A_1$
- Se $k > t+1$, seleziona elemento di rango $k-t-1$ in $A_2$

Es. $A_0 =$ | 3 | -1 | 5 | 7 | 9 | -2 | 8 | 10 | 1 | -3 | $k = 5$

$A_0 =$ | 3 | -1 | 5 | 7 | -3 | -2 | 1 | 8 | 10 | 9 | $k = 5$

$A_1$

$A_1 =$ | -3 | -2 | 5 | 7 | -1 | 3 | 1 | $k = 5$

$A_2$

$A_2 =$ | 5 | 7 | -1 | 3 | 1 | $k = 3$

$A_3 =$ | 1 | -1 | 3 | 7 | 5 | $k = 3$

Restituisce 3

COMPLESSITÁ: Ogni riorganizzazione dell'array costa $\Theta(n)$ passi

Poi viene effettuata una chiamata ricorsiva in uno delle due parti

- Nel caso pessimo la parte in cui andiamo ha lunghezza $n-1$
  Se capitasse sempre così avremmo complessità

$$T(n) = T(n-1) + \Theta(n) \longrightarrow T(n) = \Theta(n^2)$$

- Se succede che la parte in cui andiamo a cercare ha sempre lunghezza $\leq c \cdot n$ con $c$ una costante $< 1$ indipendente da $n$ e dal passo algoritmo

allora $T(n) \leq n + cn + c^2 n + \cdots$

$$\leq n \cdot \sum_{i=0}^{\infty} c^i = n \cdot \frac{1}{1-c} = \Theta(n)$$

**OSS** Se il pivot è SCELTO A CASO, allora con ALTA PROBABILITÀ il tempo di esecuzione è effettivamente $\Theta(n)$

IL CASO È MOLTO SIMILE A QUICKSORT:
- CASO PESSIMO: divisione sbilanciata
- CASO MEDIO: divisione abbastanza spesso bilanciata

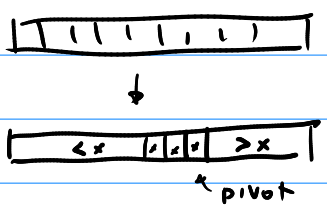
## Esiste un algoritmo per la selezione

- NON RANDOMIZZATO
- con complessità $\Theta(n)$

Va detto che la costante nascosta da $\Theta$ è abbastanza grande, e quindi IN PRATICA l'algoritmo randomizzato è altrettanto efficiente ed è un po' più semplice da realizzare.

## COSE SULLA SELEZIONE CHE NON VEDIAMO NEL CORSO

- L'analisi del caso medio quando il pivot è preso a caso
- La procedura di riorganizzazione in base al pivot (dovreste conoscerla dal quicksort)



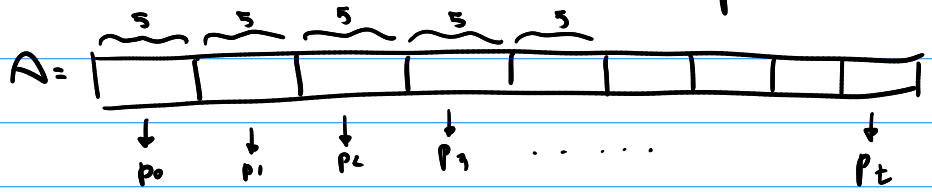
# ALGORITMO NON RANDOMIZZATO (Cap. 9.3)



- L'algoritmo non randomizzato è simile a quello randomizzato, ma la scelta del perno è più sofisticata

ALGORITMO RANDOMIZZATO : Scegli il perno a caso UNIFORMEMENTE tra gli elementi di A

ALGORITMO DETERMINISTICO : Scelta del perno :




con $t = \lfloor \frac{n}{5} \rfloor - 1$

**Passo 1** : Dividi A in segmenti di lunghezza 5 e prendi il mediano di ognuno

complessità $\Theta(n)$

**Passo 2** $p \leftarrow$ MEDIANO $([p_0, p_1, \dots, p_t])$

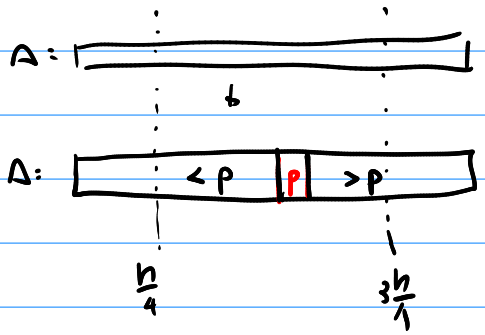
Prendiamo p: il mediano della sequenza dei mediani, come PIVOT

**Passo 3** PROSEGUIAMO L'ALGORITMO COME PRIMA

**Thm**

Se $n > 120$, il pivot p preso in questo modo garantisce che, siano

$$L = \{x \in A,\ x < p\}$$
$$H = \{x \in A,\ x > p\}$$




abbiamo $\frac{n}{4} \leq |L|, |H| \leq \frac{3n}{4}$

**Corollario**

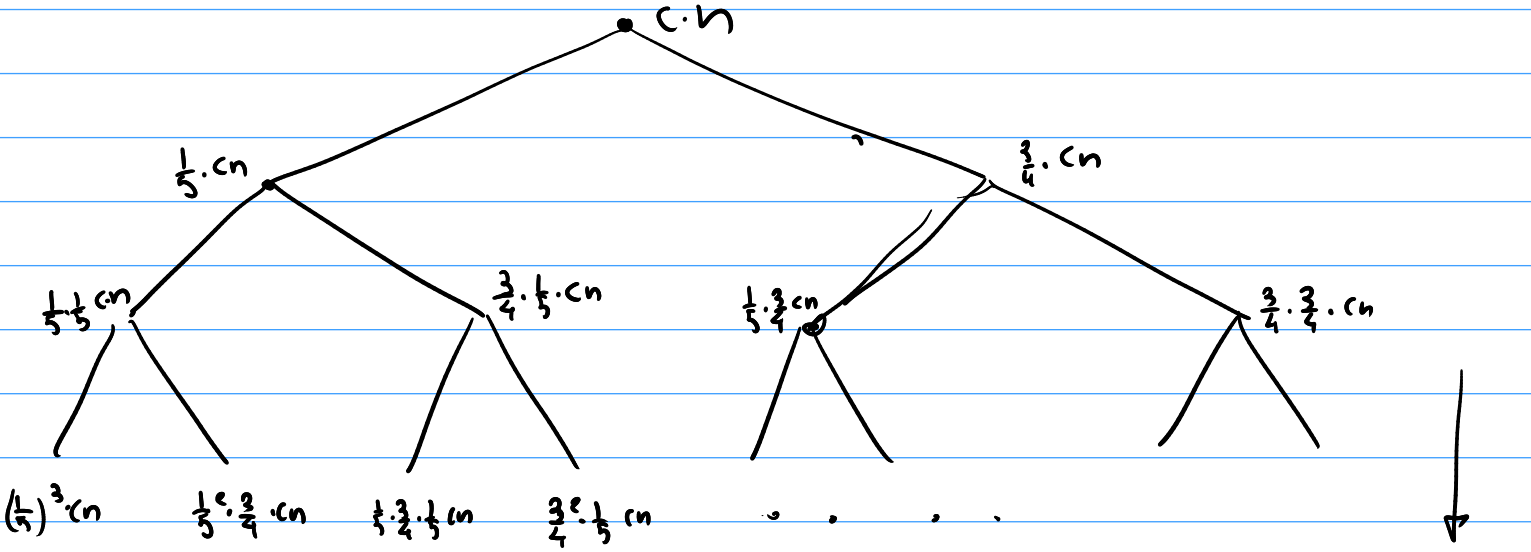
$$T(n) = \begin{cases} \Theta(1) & \text{per } n < 120 \\ T(\frac{n}{5}) + T(\frac{3}{4}n) + \Theta(n) \end{cases}$$

$\Rightarrow T(n) = \Theta(n)$

**Dim** (corollario) Sia $c$ grande abbastanza per cui

$$T(n) \leq T\left(\frac{n}{5}\right) + T\left(\frac{3}{4}n\right) + c \cdot n$$

altezza il costo è rappresentato dall'albero di ricorrenza




il costo totale è $\leq cn \cdot \sum_{i=0}^{\infty} \left(\frac{1}{5} + \frac{3}{4}\right)^i = c \cdot n \cdot \frac{1}{1 - \frac{19}{20}} = 20 \cdot c \cdot n = \Theta(n)$

Resta ora mostrare che in effetti il perno $p$ scelto in questo modo fornisce una divisione bilanciata

$$\frac{n}{4} \leq |L|, |H| \leq \frac{3}{4}n$$

**Claim (theorema)**

- La lista dei mediani ha lunghezza $\lceil \frac{n}{5} \rceil$
- $p$ è il mediano di essa e quindi
  ci sono $\left\lceil \frac{\lceil n/5 \rceil}{2} \right\rceil - 1$ mediani più piccoli di $p$ — tranne 1!
- di questi $\lceil \frac{n}{10} \rceil - 1$ mediani più piccoli di $p$, ognuno è in un gruppo di 5 con altri 2 elementi più piccoli.
  abbiamo allora $3\left(\lceil \frac{n}{10} \rceil - 2\right)$ elementi $< p$

$$\longrightarrow \frac{3}{10}n - 6 \longrightarrow |L| \geq \frac{3}{10}n - 6 \xrightarrow[n \geq 120]{} |L| > \frac{1}{4}n$$

$$|H| \leq n-1-\frac{3}{10}n+6 = \frac{7}{10}+5$$

(7)

- Analogamente $|H| \geq \frac{3}{10}n-6 \xrightarrow[n\geq 120]{} |H| \geq \frac{1}{4}n$
- Quindi $|L|, |H| \leq \frac{3}{4}n$ □

## ESERCIZIO (9.3-3)

- Spiegate come implementare quicksort in modo che abbia complessità $O(n \log n)$ NEL CASO PEGGIORE
- assumendo che tutti gli elementi siano distinti

## ESERCIZIO (9.3-1)

Che succede nell'analisi dell'algoritmo di selezione deterministico se invece di raggruppare in blocchi da 5 lo facciamo in

blocchi da 3? blocchi da 7?

## ESERCIZIO (9.3-8)

- Supponete di avere due array A, B di lunghezza n entrambi ordinati, senza elementi che compaiano più di una volta
- Scrivere un algoritmo di complessità $\Theta(\log n)$

che restituisca l'elemento di rango n tra l'unione degli elementi di A e B