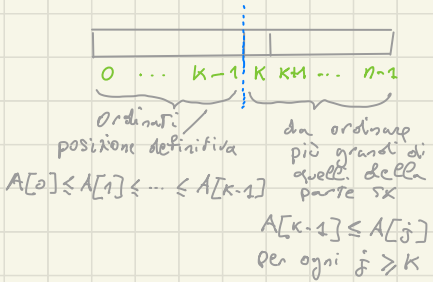
Algoritmi e Strutture Dati

Lezione 8

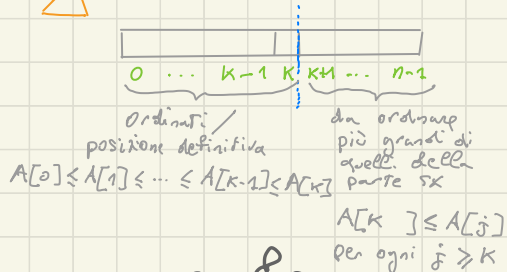
14 ottobre 2022

SelectionSort

Array A[0..n-1]



PASSO K
(k = 0..n-2)



ALGORITMO selectionSort (array A[0..n-1])

```
FOR k ← 0 TO n-2 DO
    // ricerca la posizione del minimo in A[k..n-1]
    m ← k
    FOR j ← k+1 TO n-1 DO
        IF A[j] < A[m] THEN
            m ← j
    scambia A[m] con A[k]
```

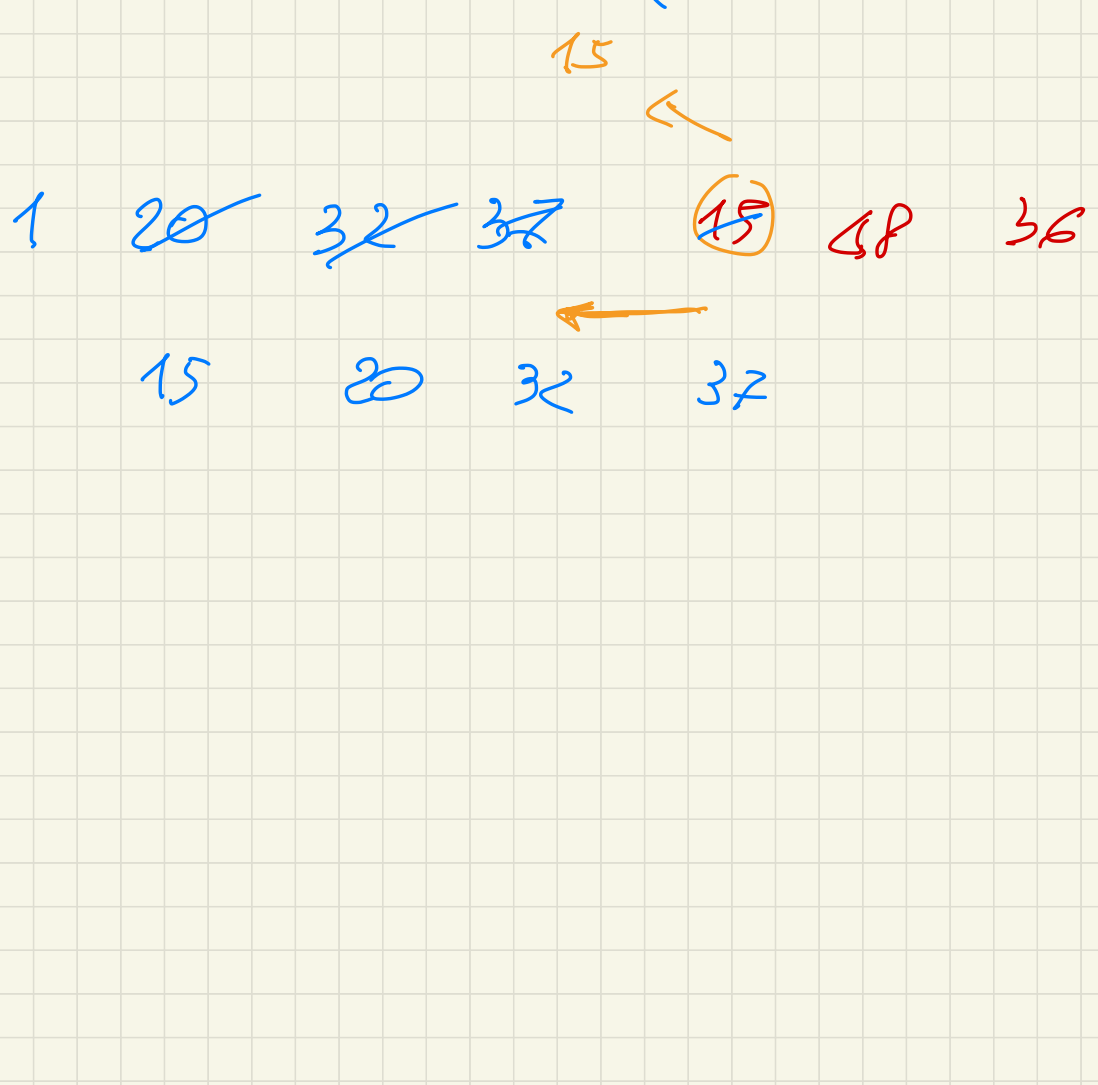
1 4 2 8

4 3 7 8 1 10

$$n-1 + n-2 + \dots + 1 = \frac{n(n-1)}{2}$$

20 20 10

10 20


15
1 20 32 37 15 48 36
15 20 32 37

InsertionSort

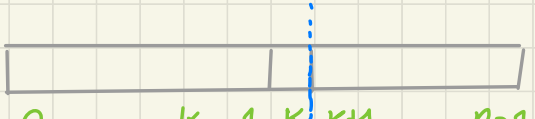
Array $A[0..n-1]$

ordinati tra loro

$A[0] \leq A[1] \leq \dots \leq A[k-1]$

PASSO k

$(k = 1, \dots, n-1)$

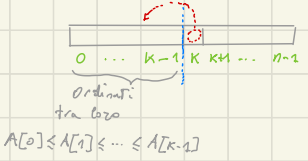


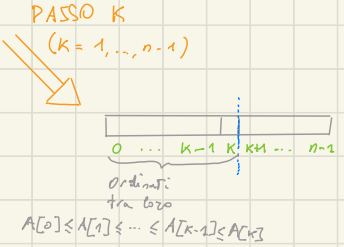
ordinati tra loro

$A[0] \leq A[1] \leq \dots \leq A[k-1] \leq A[k]$

InsertionSort

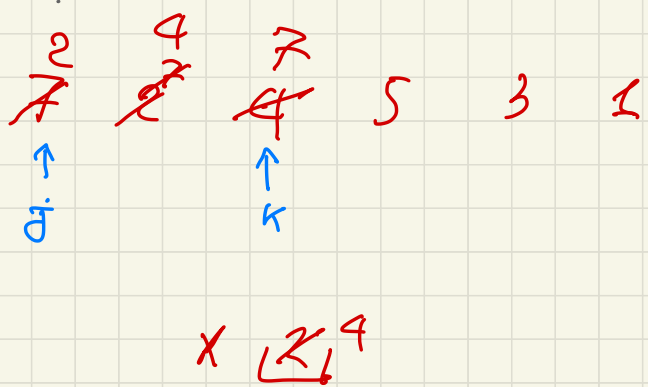
```
ALGORITMO insertionSort(array A[0..n-1])
    FOR k ← 1 TO n-1 DO
        x ← A[k]
        j ← k-1
        WHILE j ≥ 0 AND A[j] > x DO
            A[j+1] ← A[j]
            j ← j-1
        A[j+1] ← x
```

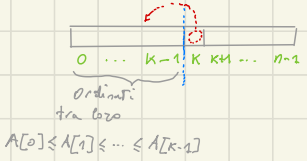


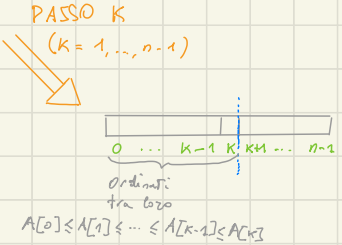


InsertionSort

```
ALGORITMO insertionSort(array A[0..n-1])
  FOR k ← 1 TO n-1 DO
      x ← A[k]
      j ← k-1
      WHILE j ≥ 0 AND A[j] > x DO
          A[j+1] ← A[j]
          j ← j-1
      A[j+1] ← x
```







InsertionSort: n° di confronti

```
ALGORITMO insertionSort(array A[0..n-1])
  FOR k ← 1 TO n-1 DO
      x ← A[k]
      j ← k-1
      WHILE j ≥ 0 AND A[j] > x DO
          A[j+1] ← A[j]
          j ← j-1
      A[j+1] ← x
```

cfr

iterazione k al più k cfr

$$\sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}$$

n° cfr nel caso peggiore

7 5 3 2 1

cfr $\Theta(n^2)$

minimo di cfr $n-1$

1 2 3 7 10

--- 4 --- 4 ---

BubbleSort

Array A[0..n-1]

```
ALGORITMO bubbleSort (Array A[0..n-1])
    DO
    |   scambiato ← false
    |   FOR j ← 1 TO n-1 DO
    |   |   IF A[j] < A[j-1] THEN
    |   |   |   scambia A[j] e A[j-1]
    |   |   |   scambiato ← true
    WHILE scambiato
```

BubbleSort

```
ALGORITMO bubbleSort (Array A[0..n-1])
    DO
    |   scambiato ← false
    |   FOR j ← 1 TO n-1 DO
    |   |   IF A[j] < A[j-1] THEN
    |   |   |   scambia A[j] e A[j-1]
    |   |   |   scambiato ← true
    |
    WHILE scambiato
```

0	1	2	3	4	5
7	2	4	5	3	1
2	4	5	3	1	7
2	4	3	1	5	7
2	3	1	4	5	7
2	1	3	4	5	7
1	2	3	4	5	7

```
ALGORITMO bubbleSort (Array A[0..n-1])
    i ← 1
    DO
        scambiato ← false
        FOR j ← 1 TO n-i DO
            IF A[j] < A[j-1] THEN
                scambia A[j] e A[j-1]
                scambiato ← true
        i ← i+1
    WHILE scambiato AND i < n
```

```
ALGORITMO bubbleSort (Array A[0..n-1])
    DO
        scambiato ← false
        FOR j ← 1 TO n-1 DO
            IF A[j] < A[j-1] THEN
                scambia A[j] e A[j-1]
                scambiato ← true
    WHILE scambiato
```

iterare i

$\downarrow$

$n-i$ cfr

$$\sum_{i=1}^{n-1}(n-i) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$$

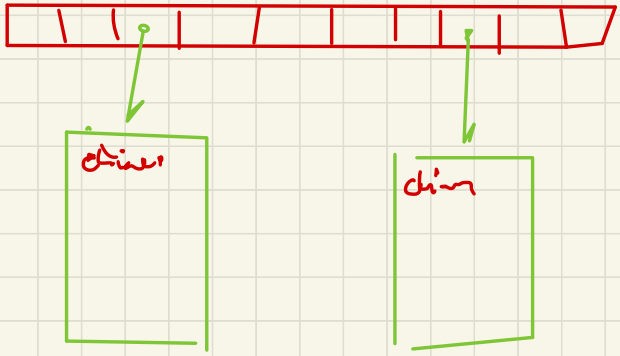
caso peggiore $\Theta(n^2)$ cfr

$n-1$ caso migliore

12 3 4 9 10

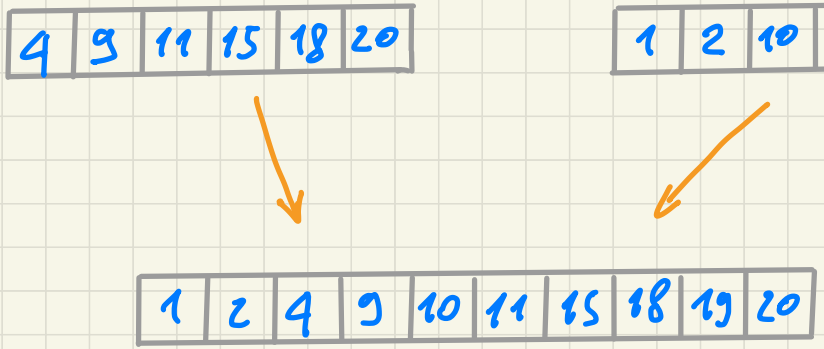
3 4 9 10 12

4 9 10 12 3



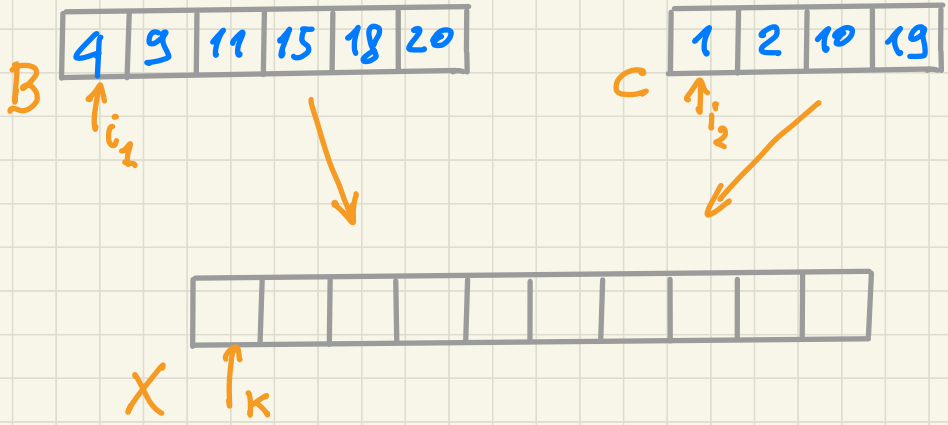
MERGE o FUSIONE

- B, C array ordinati in modo non decrescente
- Costruire un array X ordinato contenente tutti gli elementi di B e C



MERGE o FUSIONE

- B, C array ordinati in modo non decrescente
- Costruire un array X ordinato contenente tutti gli elementi di B e C



ALGORITMO merge (array $B[0..\ell_B-1]$, array $C[0..\ell_C-1]$) → array

Sia $X[0..\ell_B+\ell_C-1]$ un array

$i_1 \leftarrow 0$, $i_2 \leftarrow 0$, $K \leftarrow 0$

WHILE $i_1 < \ell_B$ AND $i_2 < \ell_C$ DO

  IF $B[i_1] \leq C[i_2]$ THEN

    $X[K] \leftarrow B[i_1]$

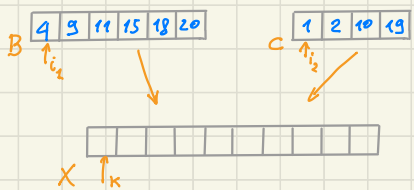
    $i_1 \leftarrow i_1 + 1$

  ELSE

    $X[K] \leftarrow C[i_2]$

    $i_2 \leftarrow i_2 + 1$

  $K \leftarrow K+1$



IF $i_1 < \ell_B$ THEN // in B è restato qualcosa

  FOR $j \leftarrow i_1$ TO $\ell_B - 1$ DO

    $X[K] \leftarrow B[j]$

    $K \leftarrow K+1$

ELSE // in C è restato qualcosa

  FOR $j \leftarrow i_2$ TO $\ell_C - 1$ DO

    $X[K] \leftarrow C[j]$

    $K \leftarrow K+1$

RETURN X

ALGORITMO merge (array $B[0..\ell_B-1]$, array $C[0..\ell_C-1]$) → array

Sia $X[0..\ell_B+\ell_C-1]$ un array

$i_1 \leftarrow 0$, $i_2 \leftarrow 0$, $k \leftarrow 0$

WHILE $i_1 < \ell_B$ AND $i_2 < \ell_C$ DO

  IF $B[i_1] \leq C[i_2]$ THEN

    $X[k] \leftarrow B[i_1]$

    $i_1 \leftarrow i_1 + 1$

  ELSE

    $X[k] \leftarrow C[i_2]$

    $i_2 \leftarrow i_2 + 1$

  $k \leftarrow k+1$

IF $i_1 < \ell_B$ THEN // in B è restato qualcosa

  FOR $j \leftarrow i_1$ TO $\ell_B - 1$ DO

    $X[k] \leftarrow B[j]$

    $k \leftarrow k+1$

ELSE // in C è restato qualcosa

  FOR $j \leftarrow i_2$ TO $\ell_C - 1$ DO

    $X[k] \leftarrow C[j]$

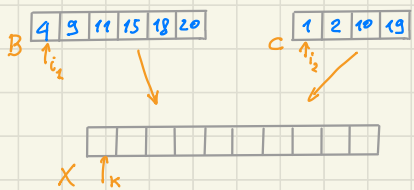
    $k \leftarrow k+1$

RETURN X

n° di cfr

$n = \ell_B + \ell_C$

= # tot elementi



1 cfr → sistemo 1 elemento in X

al max n cfr

$n - 1$

$C_{merge}(n) = n - 1$

$$\sum_{i=0}^{k-1} 2^i = 2^k - 1$$

$$\begin{array}{r}
1 \\
10 \\
100 \\
1000 \\
\cdots \\
1000\ 0\ 0\ 0 \\
\hline
1\ 1\ 1\ 1\ 1\ 1
\end{array}$$

dato n trovare N potenza di 2 t.c. $n \leq N$ N più piccolo possibile

$n = 26 = (11010)_2$

$n \cdot 2$ $\quad 1\ \cancel{1}0\ \cancel{1}00$

$\quad\quad\quad\quad\ \ 0\ \ 0$

Per ogni $n \in \mathbb{N}$ $\exists$ potenza di 2, N, t.c.

$$n \leq N < 2n$$