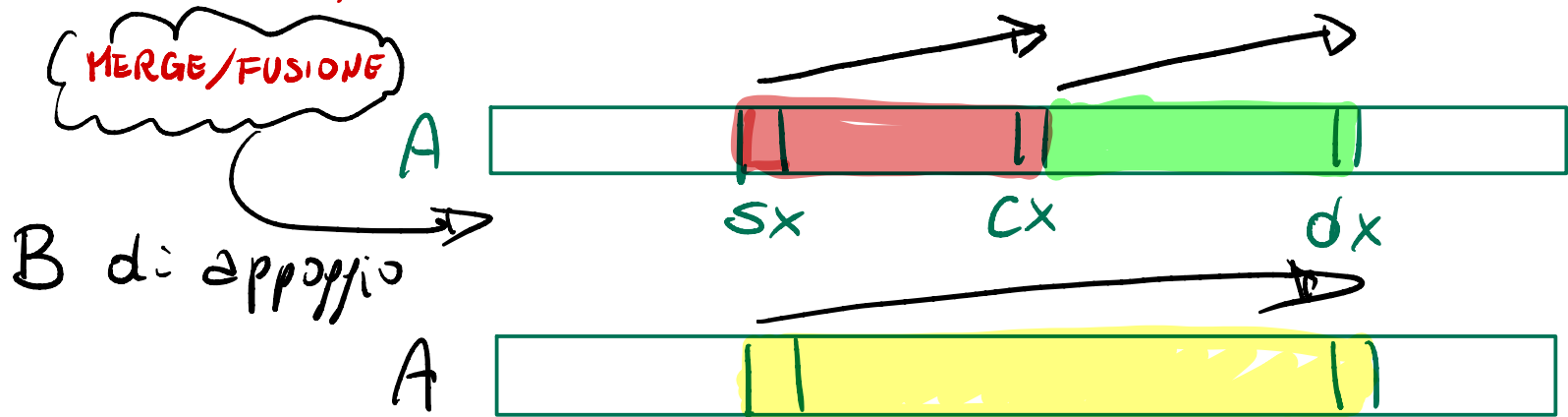


MERGESORT

- opera mediante confronti: $O(n \lg n)$ per n elementi: caso pessimo
- OTTIMO perché vale il limite inferiore $\Omega(n \lg n)$ al caso pessimo
- Molto utilizzato per enormi moli di dati (per esempio su memoria esterna)
- classico esempio di paradigma DIVIDE-ET-IMPERA

Idea: sfruttare la transitività della relazione di ordinamento



```

1  Fusione( a, sx, cx, dx ):                                (pre:  $0 \leq sx \leq cx < dx \leq n-1$ )
2      i = sx;
3      j = cx+1;
4      k = 0;
5      WHILE ((i <= cx) && (j <= dx)) {
6          IF (a[i] <= a[j]) {
7              b[k] = a[i];
8              i = i+1;
9          } ELSE {
10             b[k] = a[j];
11             j = j+1;
12         }
13         k = k+1;
14     }
15     FOR ( ; i <= cx; i = i+1, k = k+1)
16         b[k] = a[i];
17     FOR ( ; j <= dx; j = j+1, k = k+1)
18         b[k] = a[j];
19     FOR (i = sx; i <= dx; i = i+1)
20         a[i] = b[i-sx];

```

Idea: sfruttare la transitività della relazione di ordinamento

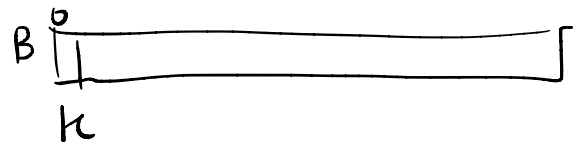
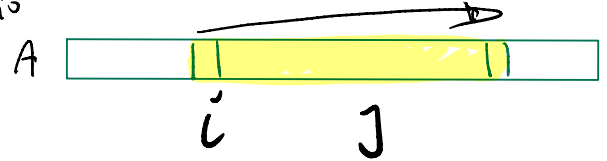
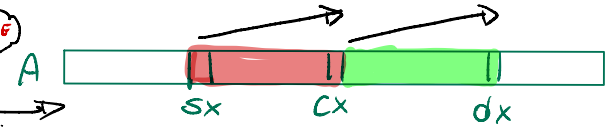
MERGE/FUSIONE

B di appoggio

ogni iterazione richiede tempo costante

ogni iterazione sicuramente incrementa k

$O(n)$ per copiare da B in A



$$n = dx - sx + 1$$

Uno solo eseguito

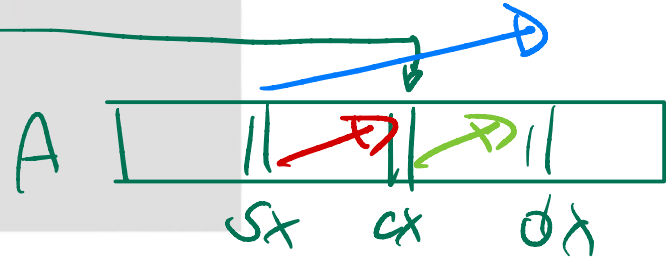
k può essere incrementato al più n volte \Rightarrow

costo totale di MERGE/FUSIONE è $O(n)$

```

1 MergeSort( a, sinistra, destra ):
2                                     <pre:  $0 \leq sinistra \leq destra \leq n - 1$ >
3 IF (sinistra < destra) {
4     centro = (sinistra+destra)/2;
5     MergeSort( a, sinistra, centro );
6     MergeSort( a, centro+1, destra );
7     Fusione( a, sinistra, centro, destra );
8 }

```



Caso base = vuoto o 1 elemento

$$\begin{matrix} sx=5 \\ dx=6 \end{matrix} \rightarrow cx=5$$

Paso induttivo = dividi $A[sx \dots dx]$ in due metà

- ordine ricorsivamente $A[sx \dots cx]$
- " " " $A[cx+1 \dots dx]$

- MERGE/FUSIONE di queste due metà

$$\begin{matrix} sx \dots cx & cx+1 \dots dx \\ 5 & 5 & 6 & 6 \\ sx \dots cx-1 & cx \dots dx \\ 5 & 4 & 5 & 6 \end{matrix}$$

```

1  Fusione( a, sx, cx, dx ):                                ⟨pre: 0 ≤ sx ≤ cx < dx ≤ n - 1⟩
2      i = sx;
3      j = cx+1;
4      k = 0;
5      WHILE ((i ≤ cx) && (j ≤ dx)) {
6          IF (a[i] ≤ a[j]) {
7              b[k] = a[i];
8              i = i+1;
9          } ELSE {
10             b[k] = a[j];
11             j = j+1;
12         }
13         k = k+1;
14     }

```

$$a[i] = \min \{ x \in A[i..cx] \cup A[j..dx] \}$$

Albero delle chiamate ricorsive $n=11$

MERGE SORT(a, s, d)
 $\uparrow \quad \uparrow$

