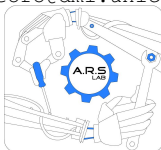


Algoritmi di Ordinamento

Corrado Santoro

Dipartimento di Matematica e Informatica

santoro@dmi.unict.it

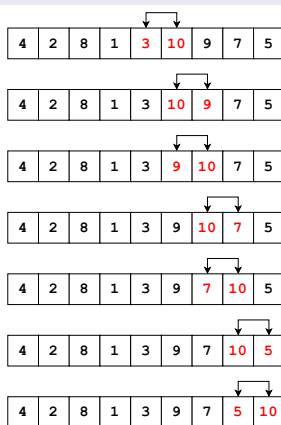
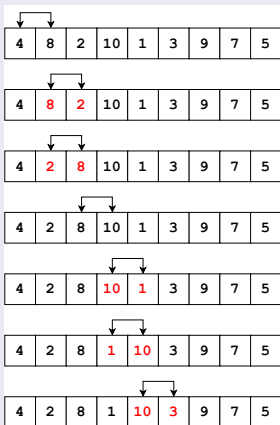


Corso di Programmazione II

- Bubble sort
- Selection (select) sort
- Insertion (insert) sort
- Quick sort
- Merge sort

Bubble Sort

- Per ogni iterazione si controllano gli elementi adiacenti
- Se l'ordine è errato si scambiano di posto
- Si ripetono i passi precedenti n volte (o fin quando non ci si accorge che l'array è ordinato)



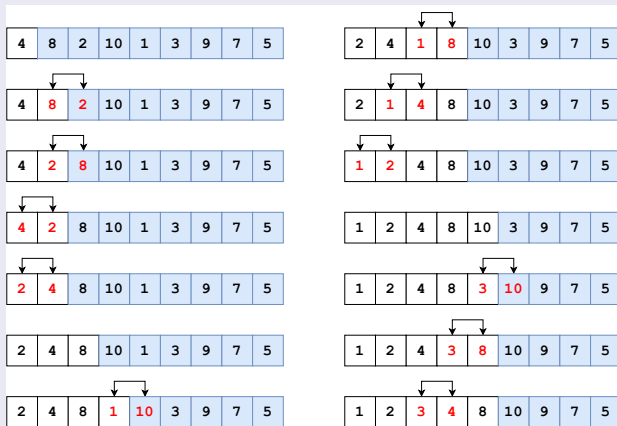
Select Sort

- Si basa sulla divisione del vettore in 2 sottovettori $v_1 = [0, i]$ e $v_2 = [i + 1, n - 1]$, il primo ordinato, il secondo no
- Per ogni iterazione si cerca l'elemento più piccolo in v_2 e lo si pone in coda a v_1



Insert Sort

- Si basa sulla divisione del vettore in 2 sottovettori $v_1 = [0, i]$ e $v_2 = [i + 1, n - 1]$, il primo ordinato, il secondo no
- Per ogni iterazione si prende il primo elemento di v_2 e lo si pone al posto giusto in v_1



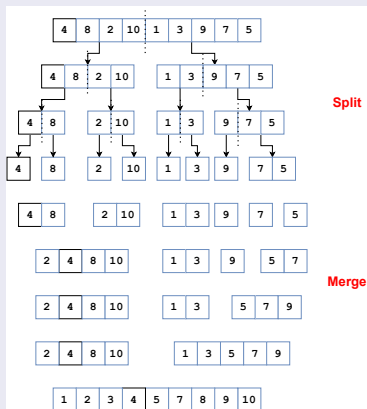
Quick Sort

- Si calcola il max tra i primi due elementi dell'array $M = \max\{v[0], v[1]\}$ e si costruiscono due sottoarray $v_1 = \{x \in v : x < M\}$ e $v_2 = \{x \in v : x > M\}$
- Si ordinano i sottovettori v_1 e v_2
- Si costruisce il vettore finale $v = v_1 + \{M\} + v_2$



Merge Sort

- Si basa sulla divisione del vettore in 2 sottovettori $v_1 = [0, n/2 - 1]$ e $v_2 = [n/2, n - 1]$
- Si ordinano i sottovettori v_1 e v_2
- Si “fondono” i due sottovettori in un unico vettore ordinato



Algoritmi di Ordinamento

Corrado Santoro

Dipartimento di Matematica e Informatica

santoro@dmi.unict.it



Corso di Programmazione II