

Esercizi

3. Algoritmi di ordinamento

Esercizio 3.1

Considerate la seguente sequenza di numeri

2222 4512 27 890 345 1234 9876 1236 2345 231 234 543

memorizzata in un array.

- Supponete di ordinare la sequenza mediante l'algoritmo di ordinamento per selezione. Indicate la sequenza ottenuta al termine di ciascuna iterazione del ciclo principale dell'algoritmo.
- Supponete di ordinare la sequenza mediante l'algoritmo di ordinamento per inserimento. Indicate la sequenza ottenuta al termine di ciascuna iterazione del ciclo principale dell'algoritmo.
- Supponete di ordinare la sequenza mediante l'algoritmo `bubbleSort`. Indicate la sequenza ottenuta al termine di ciascuna iterazione del ciclo principale dell'algoritmo.
- Supponete di ordinare la sequenza mediante l'algoritmo `radixSort`. Indicate la sequenza ottenuta al termine di ciascuna iterazione del ciclo principale dell'algoritmo.
- Supponete di ordinare la sequenza mediante l'algoritmo `heapSort`. Indicate lo heap che viene ottenuto a partire dalla sequenza, e la sequenza ottenuta al termine di ciascuna iterazione del ciclo principale dell'algoritmo.
- Supponete di ordinare la sequenza mediante l'algoritmo `quickSort`, scegliendo come perno il primo elemento. Indicate il contenuto dell'array dopo avere effettuato la partizione, prima delle chiamate ricorsive di `quickSort`.
- Supponete di ordinare la sequenza mediante l'algoritmo `mergeSort`. Indicate come viene scomposto e ricomposto l'array ad ogni livello di ricorsione.
- Costruite un albero binario di ricerca inserendo, in un albero inizialmente vuoto, gli elementi della sequenza nell'ordine indicato, utilizzando la versione base dell'inserimento (senza nessun bilanciamento). Indicate poi le sequenze che si ottengono visitando l'albero in ordine anticipato, simmetrico, posticipato.
- Ripetete il punto precedente, costruendo però un albero AVL. Indicate poi le sequenze che si ottengono visitando l'albero in ordine anticipato, simmetrico, posticipato.
- Ripetete il punto precedente, costruendo però un albero 2-3.

Esercizio 3.2

Ripetete l'esercizio 3.1, considerando la sequenza di numeri

4567 8776 2324 3434 9876 123 876 1232 234 975 654 444

Esercizio 3.3

Scrivete una versione modificata di `radixSort` per trattare anche numeri negativi.

Esercizio 3.4

Scrivete una versione di `radixSort` per ordinare stringhe di lettere sull'alfabeto $\{a, \dots, z\}$. Fate attenzione al fatto che le stringhe potrebbero avere lunghezza differente.

Esercizio 3.5

Ripetete l'esercizio 3.1, considerando la sequenza contenente le parole

gatto topo papera cane formica oca pantera falco lupo zebra ape vespa

e l'ordinamento lessicografico.

Esercizio 3.6

Ripetete l'esercizio 3.1, considerando la sequenza contenente le parole

giallo rosso verde viola blu nero bianco arancio rosa lilla

e l'ordinamento lessicografico.