Strutture dati, Algoritmi e Complessità

Esercitazione 5

26 Marzo 2024

In questa esercitazione, lo scheletro del codice sarà fornito solo in JAVA. Gli studenti che intendono lavorare in C possono farlo, replicando lo schema proposto.

Esercizio 1

Scrivere una funzione permuta_negativi − positivi(float[]a) che implementi un algoritmo efficiente con le seguenti caratteristiche: dato in input un array $A[1..n]$ di $n \geq 1$ numeri reali, permuta A in modo tale che tutti i valori negativi compaiano prima di tutti i valori positivi. Non è richiesto che i valori positivi e negativi siano a loro volta ordinati. Ad esempio, se $A = [-3, 2, 4, 6, -3, -4]$, la funzione può restituire, ad esempio, il vettore $[-3, -3, -4, 2, 4, 6]$ oppure il vettore $[-3, -4, -3, 6, 2, 4]$.

Esercizio 2

Il problema seguente fu inizialmente proposto da Edsger W. Dijkstra come *Dutch National Flag Problem*; ne proponiamo una rivisitazione in chiave italiana. Sia $A[1..n]$ un array di caratteri, i cui elementi assumono valori nell'insieme 'B', 'R', 'V' ("bianco", "rosso" e "verde"). Scrivere una funzione bandiera(char[]a) che implementa un algoritmo efficiente in grado di permutare gli elementi di A in modo tale che tutti i valori 'V' precedano tutti i valori 'B', i quali a loro volta precedano tutti i valori 'R'. L'algoritmo dovrebbe richiedere spazio aggiuntivo $O(1)$, quindi dovrebbe limitarsi a permutare tra loro gli elementi di A senza usare un secondo array di appoggio.

Esercizio 3

Una matrice quadrata $M[1..n, 1..n]$ di $n \times n$ interi è un quadrato latino se ogni intero nell'insieme $\{1, ..., n\}$ compare una e una sola volta in ogni riga e in ogni colonna. Ad esempio, questa matrice rappresenta un quadrato latino:

1	2	3	4
2	1	4	3
4	3	2	1
3	4	1	2

Scrivere una funzione isQuadratoLatino(int[][]m) che implementi un algoritmo efficiente che, data in input una matrice quadrata M[1..n, 1..n] di interi arbitrari, restituisca true se e solo se M rappresenta un quadrato latino.

Esercizio 4

Si consideri un array $A[1..n]$ di n numeri reali distinti e un intero $k \in \{1, ..., n\}$; l'array non è ordinato. Scrivere ina funzione primiKMin(float[]A, intk) un algoritmo che, dati A e k, restituisca o stampi i k valori più piccoli presenti in A.