

Esame di algoritmi e strutture dati

15 febbraio 2019

Tempo a disposizione: 2 ore

Esercizio 1

(6 punti)

Progettare un algoritmo (pseudocodice) con segnatura:

Boolean CamminoPari(AlberoBinario T)

che, preso in input un albero binario T i cui nodi contengono chiavi intere, restituisce *true* se e solo se esiste un cammino dalla radice alla foglia che contiene solo valori pari.

Esercizio 2

(6 punti)

Risolvere le seguenti equazioni di ricorrenza, illustrando il procedimento seguito:

1. $T(n) = \begin{cases} nT(n-1), & \text{se } n > 1 \\ 1, & \text{se } n = 1 \end{cases}$
2. $T(n) = \begin{cases} 3T(\frac{n}{9}) + n, & \text{se } n > 1 \\ 1, & \text{se } n = 1 \end{cases}$

Esercizio 3

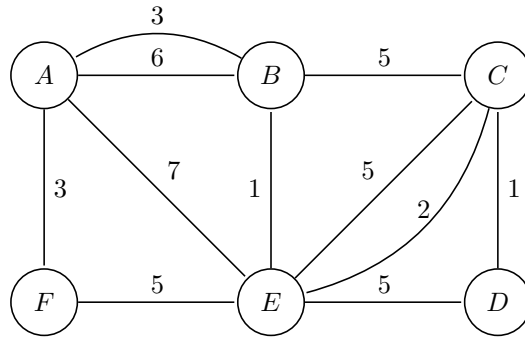
(6 punti)

1. Mostrare un algoritmo (pseudocodice) che permetta di verificare se un grafo non orientato è connesso.
2. Discutere la complessità dell'algoritmo presentato.

Esercizio 4

(5 punti)

1. Fornire la definizione di *minimo albero ricoprente*.
2. Fornire la definizione di *albero dei cammini minimi*.
3. Dato il grafo sotto riportato, mostrarne un minimo albero ricoprente ed un albero dei cammini minimi.



Esercizio 5

(7 punti)

1. Mostrare lo pseudocodice dell'algoritmo MergeSort.
2. Dimostrare formalmente il costo temporale dell'algoritmo nel caso peggiore.
3. Mostrare i passi eseguiti dall'algoritmo sul seguente array di input:

18	67	2	15	12	10	8	1
----	----	---	----	----	----	---	---

Esercizio 6

(3 punti)

ATTENZIONE: esercizio valido solo se con gli esercizi 1–5 viene raggiunto il punteggio minimo di 20.

Descrivere, con un semplice esempio, le principali tecniche di risoluzione dei conflitti nelle hash table.