Algoritmi e Strutture Dati

a.a. 2018/19

Compito del 13/01/2020

Cognome: _____ Nome: _____

Matricola: _____ E-mail: _____

Parte I

(30 minuti; ogni esercizio vale 2 punti)

1. Sia T un albero binario di ricerca contenente n nodi, sia k una chiave che occorre in T e sia x un nodo di T. Si indichi quali tra le seguenti affermazioni sono corrette, motivando brevemente le risposte:
   a. la ricerca di k in T ha costo $O(n)$;
   b. il predecessore di x si trova nel sottoalbero radicato in *x.left*;
   c. la chiave massima si trova in una foglia;
   d. eseguendo una visita posticipata di T le chiavi vengono stampate in ordine decrescente.

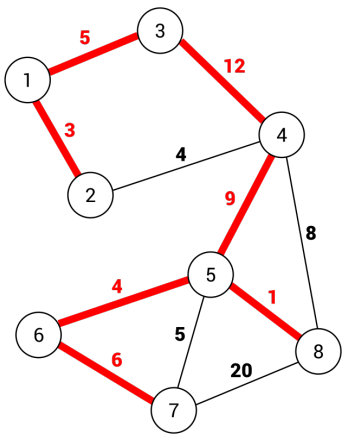
2. Per un certo problema sono stati trovati due algoritmi risolutivi (A_1 e A_2) con i seguenti tempi di esecuzione (dove n rappresenta la dimensione dei dati di ingresso):

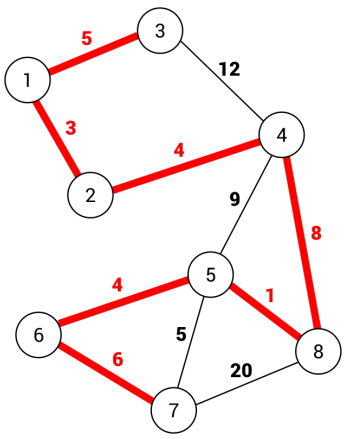
   A_1: $T(n) = 8n/2 + n^2 \log n$

   A_2: $T(n) = 8T(n/2) + n^2 \log n$

   Si dica, giustificando la risposta, quale dei due algoritmi è preferibile per n sufficientemente grande.

3. Si dica, **giustificando tecnicamente la risposta**, se nei grafi sottostanti gli archi indicati in grassetto formano o meno (1) un albero di copertura; (2) un albero di copertura minimo:





Algoritmi e Strutture Dati

a.a. 2018/19

Compito del 13/01/2020

Cognome: _____ Nome: _____

Matricola: _____ E-mail: _____

Parte II

(2.5 ore; ogni esercizio vale 6 punti)

1. a) Dare la definizione di **antenato** di un nodo in un albero *T*.

   b) Sia *T* un albero binario di ricerca contenente *n* chiavi **distinte**. Siano inoltre k_1, k_2 due chiavi contenute in *T*. Scrivere una funzione **efficiente** *antenato(Tree t, int* k_1*, int* k_2*)* che restituisce *1* se k_1 è un antenato di k_2 in *T*, *0* altrimenti. **Si devono scrivere le eventuali funzioni/procedure ausiliarie utilizzate.**

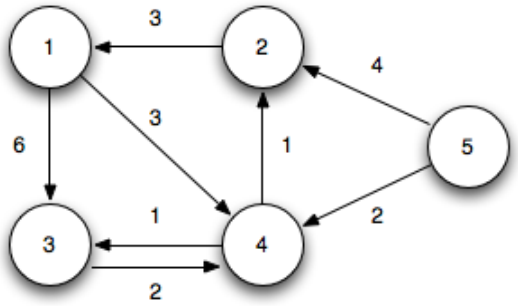
   c) Determinare e giustificare la complessità della soluzione proposta.

   Per l'esame da **12 CFU**, deve essere fornita **una funzione C**.
   Per l'esame da **9 CFU**, è sufficiente specificare lo pseudocodice.

2. Realizzare una funzione **efficiente** *triplo* che, dato un array *A* di *n* interi, verifica se esiste una coppia di indici *i, j* tali che $A[j] = 3 * A[i]$. Restituisce 1 e i corrispondenti indici se la coppia esiste, 0 altrimenti.

   Analizzare la complessità e **scrivere le eventuali funzioni/procedure ausiliarie utilizzate.**

3. Si scriva l'algoritmo di Floyd-Warshall per determinare i cammini minimi tra tutte le coppie di vertici in un grafo, si derivi la sua complessità, e si simuli la sua esecuzione sul seguente grafo:



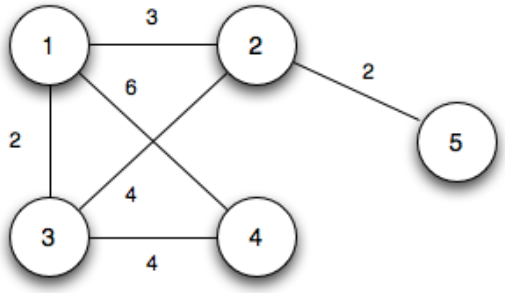
   Si stabilisca inoltre se le seguenti affermazioni sono vere o false (fornendo una dimostrazione nel primo caso e un controesempio nel secondo):

   a) "Se la diagonale principale della matrice *W* restituita in uscita dall'algoritmo di Floyd-Warshall contiene un elemento $w_{ii} < 0$, allora il grafo dato in ingresso contiene un ciclo negativo".

   b) "Se il grafo contiene un ciclo positivo, allora la diagonale principale della matrice *W* restituita in uscita dall'algoritmo contiene un elemento $w_{ii} > 0$".

   Nota: si giustifichino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative l'esercizio non verrà valutato pienamente.

4. Si scriva l'algoritmo di Prim, si dimostri la sua correttezza, si fornisca la sua complessità computazionale e si simuli accuratamente la sua esecuzione sul seguente grafo (utilizzando il vertice 1 come "sorgente"):



In particolare:

a) si indichi l'ordine con cui vengono estratti i vertici
b) si riempia la tabella seguente con i valori dei vettori key e π, iterazione per iterazione:

	vertice 1		vertice 2		vertice 3		vertice 4		vertice 5	
	key[1]	**π[1]**	**key[2]**	**π[2]**	**key[3]**	**π[3]**	**key[4]**	**π[4]**	**key[5]**	**π[5]**
dopo inizializzazione										
iterazione 1										
iterazione 2										
iterazione 3										
iterazione 4										
iterazione 5										

Nota: si giustifichino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative l'esercizio non verrà valutato pienamente.