Algoritmi e Strutture Dati

a.a. 2021/22

Compito del 6/6/2022

Cognome: _____ Nome: _____

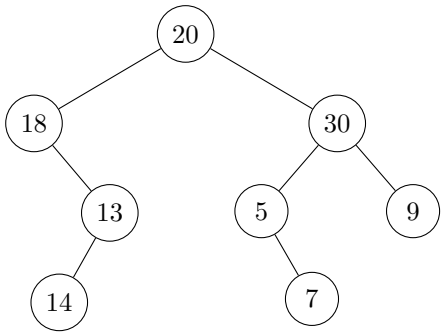
Matricola: _____ E-mail: _____

Parte I

(30 minuti; ogni esercizio vale 2 punti)

Avvertenza: Si giustifichino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative gli esercizi non verranno valutati pienamente.

1. Dato il seguente albero



   Eseguire una visita in preordine, una visita in ordine simmetrico, una visita in postordine e una visita in ampiezza elencando nei quattro casi la sequenza dei nodi incontrati.

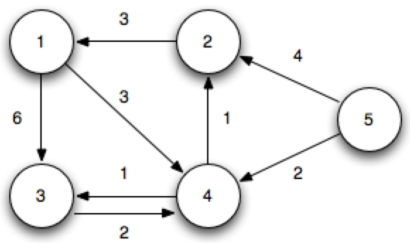
2. Un algoritmo ricorsivo $\mathcal{A}$ accetta in ingresso un grafo orientato e pesato $G = (V, E, w)$, due vertici $u, v \in V$ e una costante k, e restituisce TRUE se esiste in G un cammino tra u e v di lunghezza minore o uguale a k e FALSE in caso contrario. La sua complessità è data da

$$T(n) = 4T\left(\frac{n}{2}\right) + 3n^2$$

   dove n rappresenta il numero di vertici del grafo. Esistono algoritmi più efficienti di $\mathcal{A}$ per risolvere il problema dato? Si assuma che $w(u, v) \geq 0$ per ogni $(u, v) \in E$.

3. Si indichi l'ordine in cui l'algoritmo di Dijkstra estrae i vertici dal seguente grafo, considerando i seguenti due casi:

   a) vertice sorgente $s = 1$
   b) vertice sorgente $s = 5$



Algoritmi e Strutture Dati

a.a. 2021/22

Compito del 6/6/2022

Cognome: _____ Nome: _____

Matricola: _____ E-mail: _____

Parte II

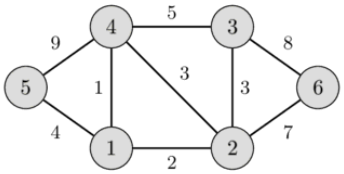
(2.5 ore; ogni esercizio vale 6 punti)

Avvertenza: Si giustifichino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative gli esercizi non verranno valutati pienamente.

1. Sia T un albero binario. Si vogliono stampare le chiavi di T memorizzate in nodi il cui sottoalbero radicato nel figlio sinistro contiene più chiavi del sottoalbero radicato nel figlio destro.
   a) Si rappresenti un **albero binario di ricerca** la cui visita in preordine ha come risultato 30, 25, 21, 40, 35, 45. Si mostri quali chiavi verrebbero stampate in base alla condizione sopra descritta.
   b) Scrivere una procedura **efficiente** in C o C++ per risolvere il problema proposto.
   c) Valutarne la complessità.

2. Scrivere un algoritmo per individuare, all'interno di una stringa $x_1 \ldots x_n$ la lunghezza massima di una sottostringa (di caratteri consecutivi) palindroma. Ad esempio, nella stringa *colonna* la sottostringa palindroma di lunghezza massima è *olo*, dunque la lunghezza massima è *3*. Più precisamente:
   a) dare una caratterizzazione ricorsiva della lunghezza massima *lung[i, j]* di una sottostringa palindroma di $x_i \ldots x_j$;
   b) tradurre tale definizione in un algoritmo di programmazione dinamica con il metodo **top-down** che determina la lunghezza massima;
   c) valutare e giustificare la complessità dell'algoritmo

3. Si scriva l'algoritmo di Prim, si dimostri la sua correttezza (ovvero: si enunci e si dimostri il teorema fondamentale degli alberi di copertura minimi e si spieghi come questo garantisca la correttezza dell'algoritmo), si fornisca la sua complessità computazionale e si simuli accuratamente la sua esecuzione sul seguente grafo utilizzando il vertice 1 come "sorgente":



In particolare: si indichi l'ordine con cui vengono estratti i vertici e 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		vertice 6	
	key[1]	**π[1]**	**key[2]**	**π[2]**	**key[3]**	**π[3]**	**key[4]**	**π[4]**	**key[5]**	**π[5]**	**key[6]**	**π[6]**
dopo inizializzazione												
iterazione 1												
iterazione 2												
iterazione 3												
iterazione 4												
iterazione 5												
iterazione 6												

4. Sia $A = (a_{ij})$ la matrice di adiacenza di un grafo orientato $G = (V, E)$.

4.1. Cosa rappresentano gli elementi della matrice

$$A^2 = A \cdot A$$

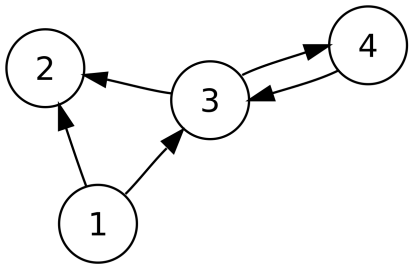
dove "·" denota l'operazione di moltiplicazione tra matrici?

4.2. Più in generale, dato un intero $k \geq 2$, cosa rappresentano gli elementi della matrice

$$A^k = \underbrace{A \cdot A \cdot \ldots \cdot A}_{k} ?$$

(Si forniscano adeguate dimostrazioni formali a supporto delle affermazioni fatte.)

4.3. Si determinino le matrici A, A^2 e A^3 per il grafo riportato di seguito:



4.4. Il seguente algoritmo accetta in ingresso la matrice di adiacenza di un grafo orientato e restituisce TRUE o FALSE. Qual è la funziona svolta e qual è la sua complessità? Giustificare le risposte.

```
MyAlgorithm(A)

1.  n = numero di righe di A
2.  crea due matrici n x n B e C

3.  for i = 1 to n
4.     for j = 1 to n
5.        B[i,j] = 0
6.        for k = 1 to n
7.           B[i,j] = B[i,j] + A[i,k]*A[k,j]

8.  for i = 1 to n
9.     for j = 1 to n
10.       C[i,j] = 0
11.       for k = 1 to n
12.          C[i,j] = C[i,j] + B[i,k]*A[k,j]

13. for i = 1 to n
14.    if C[i,i] = 0 then
15.      return TRUE

16. return FALSE
```