Algoritmi e Strutture Dati

a.a. 2020/21

Compito del 26/08/2021

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.

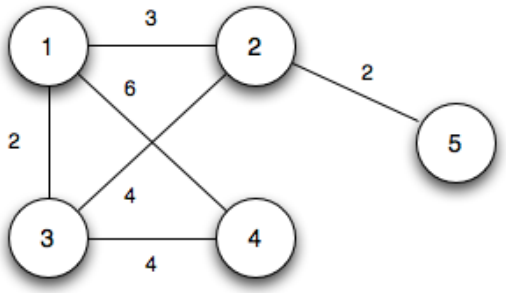
1. Scrivere l'algoritmo build-Min-Heap e simulare la sua esecuzione sull'array <35, 42, 2, -1, 27>

2. Un algoritmo ricorsivo $\mathcal{A}$ determina i cammini minimi tra tutte le coppie di vertici in un grafo pesato e ha complessità pari a:

   $$T(n) = 3T(n/2) + 3n^2$$

   dove n rappresenta il numero di vertici del grafo. Si stabilisca se $\mathcal{A}$ è asintoticamente più efficiente dell'algoritmo di Floyd-Warshall.

3. Si supponga di eseguire l'algoritmo di Prim sul seguente grafo, utilizzando il vertice 1 come sorgente:



   a) Con quale ordine verranno estratti i vertici?
   b) Si determini l'albero di copertura minimo restituito dall'algoritmo.

Algoritmi e Strutture Dati

a.a. 2020/21

Compito del 26/08/2021

Cognome: _____ Nome: _____

Matricola: _____ E-mail: _____

Parte II

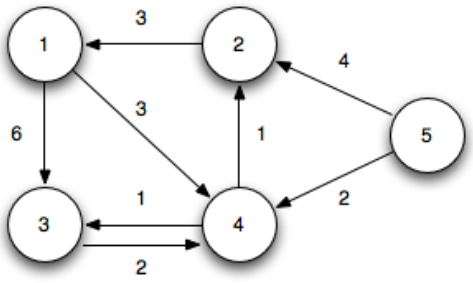
(2.5 ore; ogni esercizio vale 6 punti)

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

1. Si consideri un albero binario *T*, i cui nodi *x* hanno i campi *left, right, p* e *key* che rappresentano il figlio sinistro, il figlio destro, il padre e la chiave del nodo, rispettivamente. Un cammino è una sequenza di nodi $x_0,x_1,...,x_n$ tale che per ogni $i = 0,...,n-1$ vale $x_{i+1} \rightarrow p = x_i$. Il cammino è detto *terminabile* se $x_n \rightarrow l = NULL$ oppure $x_n \rightarrow r = NULL$. Diciamo che l'albero è *3-bilanciato* se tutti i cammini terminabili che partono dalla radice di T hanno lunghezze che differiscono al più di *3*.
   a. Scrivere una funzione **efficiente in C** *bal(u)* che dato in input la radice *u* di un albero *T* verifica se è *3-bilanciato* e ritorna *1* se *T* è *3-bilanciato*, *0* altrimenti.
   b. Valutare la complessità della funzione, **indicando eventuali relazioni di ricorrenza.**

2. Si scriva un algoritmo **efficiente** che, ricevuto in ingresso un insieme di *n* intervalli chiusi $[a_i, b_i]$, con a_i, b_i numeri interi, $a_i \le b_i$ e $1 \le i \le n$, stabilisca se la loro unione è un intervallo (nel qual caso, restituisca tale intervallo) o meno.
   Calcolare e giustificare la complessità dell'algoritmo proposto.
   Si devono scrivere le eventuali procedure/funzioni ausiliarie utilizzate.

3. Si scriva l'algoritmo di Dijkstra, si derivi la sua complessità, si dimostri la sua correttezza e si simuli 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 d e π, iterazione per iterazione:

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

4. Si definiscano le classi di complessità P, NP ed NPC e si enunci e dimostri il teorema fondamentale della NP-completezza. Si formuli inoltre il problema CLIQUE e si mostri che è NP-completo.