

Algoritmi e Strutture Dati
a.a. 2014/15

Compito del 20/01/2016

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

Parte I

(30 minuti; ogni esercizio vale 2 punti)

1. Completare la seguente tabella indicando la complessità delle operazioni che si riferiscono a un dizionario di n elementi. Per l'operazione **Successore** si assuma di essere sull'elemento x a cui si applica l'operazione.

	Ricerca	Massimo	Successore	Costruzione
Tabelle Hash con liste di collisione (caso medio)*				
Tabelle Hash a indirizzamento aperto (caso medio)*				
Tabelle Hash a indirizzamento aperto (caso pessimo)*				

*La tabella Hash ha dimensione m e il fattore di carico è α

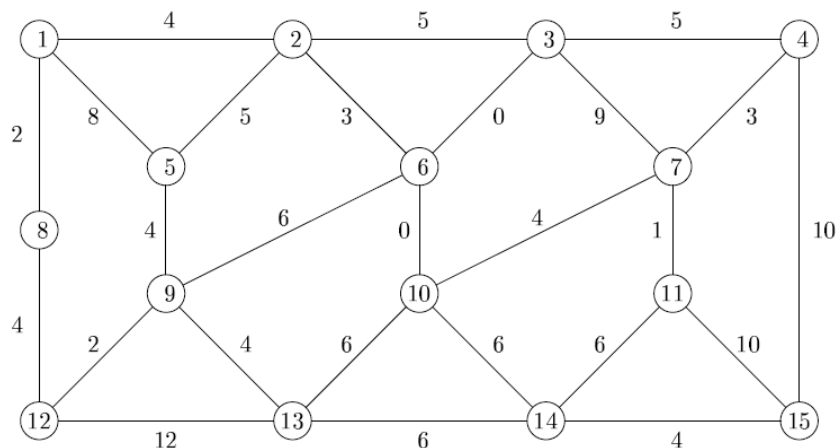
2. Per un certo problema sono stati trovati due algoritmi risolutivi (A_1 e A_2) con i seguenti tempi di esecuzione:

$$A_1: \quad T(n) = 3T(n/4) + n \log n$$

$$A_2: \quad T(n) = 3T(n/3) + n/2$$

Si dica, giustificando tecnicamente la risposta, quale dei due algoritmi è preferibile per input di dimensione sufficientemente grande.

3. Si determini un albero di copertura **massimo** nel seguente grafo (si marchino semplicemente gli archi sulla figura riportata di seguito):



Algoritmi e Strutture Dati

a.a. 2014/15

Compito del 20/01/2016

Cognome: _____

Nome: _____

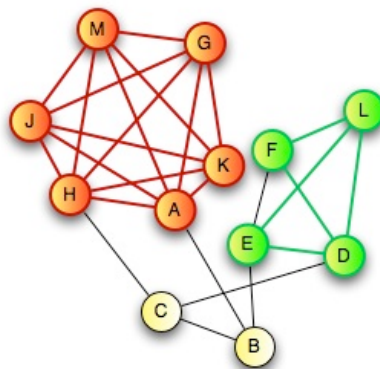
Matricola: _____

E-mail: _____

Parte II

(2.5 ore; ogni esercizio vale 6 punti)

1. Dato un albero generico i cui nodi hanno attributi **key**, **left-child**, **right-sib**, scrivere una funzione in C (**12 CFU**) o in pseudocodice (**9 CFU**) che restituisce il numero di nodi interni i cui figli hanno tutti la stessa chiave.
Qual è la complessità della funzione?
2. Sia T un albero binario di ricerca di altezza h e avente n nodi con chiavi intere eventualmente ripetute. Si progetti un algoritmo **efficiente** che, ricevuto in ingresso T e un intero k , conta il numero di occorrenze di k in T .
Analizzare la complessità dell'algoritmo.
3. Si definiscano formalmente le relazioni O , Ω , Θ e, **utilizzando le definizioni date e nient'altro**, si dimostri la verità o la falsità di ciascuna delle seguenti affermazioni:
 - a) $\log n! = O(n \log n)$
 - b) $\sqrt{n+10} = \Theta(\sqrt{n})$
 - c) $3n \log n = O(5n + \log n^3)$
 - d) $2^n = \Omega(2^{n+k})$, dove k è una costante intera positiva
 - e) $2^{n+n} = O(2^n)$
4. Si scriva un algoritmo di complessità $O(n^2)$ per determinare una clique massimale all'interno di un grafo non orientato con n vertici. Si discuta la sua complessità e la sua correttezza e si simuli accuratamente la sua esecuzione sul seguente grafo:



L'algoritmo è in grado di determinare anche clique massime? In caso negativo si fornisca un controesempio.