

# Algoritmi e Strutture Dati

a.a. 2014/15

Compito del 11/06/2015

Cognome: \_\_\_\_\_

Nome: \_\_\_\_\_

Matricola: \_\_\_\_\_

E-mail: \_\_\_\_\_

## Parte I

(30 minuti; ogni esercizio vale 2 punti)

1. Nell'ipotesi di indirizzamento aperto, scrivere uno pseudocodice per HASH-SEARCH. Quale è la complessità nel caso medio?

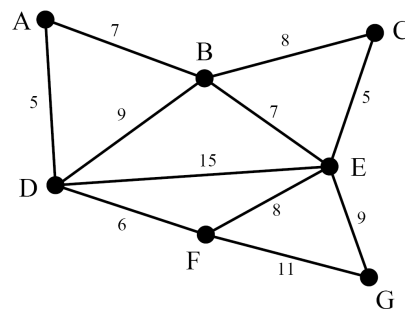
2. Per un certo problema sono stati trovati due algoritmi risolutivi ( $A_1$  e  $A_2$ ) con i seguenti tempi di esecuzione:

$$A_1: \quad T(n) = 3 \cdot T(n/2) + n^2$$

$$A_2: \quad T(n) = 4 \cdot T(n/2) + n^2$$

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 minimo nel seguente grafo:



# Algoritmi e Strutture Dati

a.a. 2014/15

## Compito del 11/06/2015

Cognome: \_\_\_\_\_

Nome: \_\_\_\_\_

Matricola: \_\_\_\_\_

E-mail: \_\_\_\_\_

### Parte II

(2.5 ore; ogni esercizio vale 6 punti)

1. Scrivere un algoritmo **efficiente** che date in input due stringhe di lunghezza  $n$  i cui caratteri variano in  $\{a, b, \dots, x, y, z\}$  rappresentate mediante due array di caratteri  $A[1..n]$  e  $B[1..n]$ , restituisce *true* se e solo se le due stringhe sono una l'anagramma dell'altra. Analizzare la complessità dell'algoritmo.

Per esempio:  $A = ['s', 'a', 'r', 'a']$  e  $B = ['r', 'a', 's', 'a']$  sono anagrammi.

$A = ['c', 'a', 'r', 'a']$  e  $B = ['r', 'a', 'n', 'a']$  NON sono anagrammi.

2. Dati due alberi binari di ricerca  $T_1$  e  $T_2$  tali che le chiavi in  $T_1$  sono tutte minori delle chiavi in  $T_2$ , scrivere una procedura che restituisce un albero di ricerca contenente tutte le chiavi in tempo  $O(h)$ , dove  $h$  è l'altezza massima dei due alberi. Quale è l'altezza dell'albero risultante?
3. Si definisca formalmente la relazione di riducibilità polinomiale tra problemi decisionali ( $\leq_P$ ) e si stabilisca se le seguenti affermazioni sono vere o false:
  - 1) La relazione  $\leq_P$  è transitiva
  - 2) La relazione  $\leq_P$  è riflessiva
  - 3) Se  $\leq_P$  è simmetrica, allora  $P = NPC$
  - 4) Se  $P \leq_P Q$  e  $Q \in P$ , allora  $P \in P$
  - 5) Se  $P, Q \in NPC$ , allora  $P \leq_P Q$  se e solo se  $Q \leq_P P$

Nel primo caso si fornisca una dimostrazione rigorosa, nel secondo un controesempio.

(**Nota:** in caso di discussioni poco formali l'esercizio non verrà valutato pienamente.)

4. Si scriva l'algoritmo di Floyd-Warshall, si dimostri la sua correttezza, si fornisca la sua complessità computazionale e si simuli accuratamente la sua esecuzione sul seguente grafo:

