

# Algoritmi e Strutture Dati

a.a. 2013/14

Compito del 28/01/2015

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

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

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

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

## Parte I

(30 minuti; ogni esercizio vale 2 punti)

1. In una tabella Hash di  $m = 17$  posizioni, inizialmente vuota, devono essere inserite le seguenti chiavi numeriche nell'ordine indicato:

18, 36, 155, 19

La tabella è a indirizzamento aperto e la scansione è eseguita per doppio Hashing:

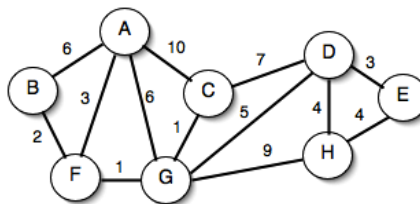
$$h(k, i) = (k \bmod m + i * 2^{k \bmod 5}) \bmod m$$

Indicare per ogni chiave le posizioni scandite nella tabella e la posizione finale dove viene allocata.

2. Si risolvano le seguenti ricorrenze:

- $T(n) = 3T(n/2) + n^2$
- $T(n) = 4T(n/2) + n^2$
- $T(n) = T(n/2) + 2^n$

3. Si determinino due diversi alberi di copertura minimi nel seguente grafo:



# Algoritmi e Strutture Dati

a.a. 2013/14

## Compito del 28/01/2015

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

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

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

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

### Parte II

(2.5 ore; ogni esercizio vale 6 punti)

1. Un nodo di un albero binario **u** è detto **intermedio** se la somma delle chiavi contenute nei nodi del sottoalbero di cui **u** è radice è uguale alla somma delle chiavi contenute nei nodi sul percorso che collega **u** alla radice dell'albero (**u** escluso).
  - a) Scrivere un algoritmo **efficiente** che restituisca il numero di nodi intermedi.
  - b) Analizzare la complessità della soluzione trovata.

Il tipo **Node** utilizzato per rappresentare l'albero binario è il seguente:

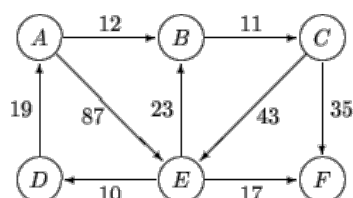
```
typedef struct node{  
    int key;  
    struct node * left;  
    struct node * right;  
} * Node;
```

Per l'esame da **12 CFU**, deve essere fornita **una funzione C**.

Per l'esame da **9 CFU**, è sufficiente specificare lo pseudocodice.

2. Scrivere un algoritmo **efficiente**, di tipo **divide et impera**, che conta il numero di occorrenze della sequenza 'a' 'r' memorizzata in posizioni adiacenti in un array di caratteri.  
Analizzare la complessità indicando e risolvendo la corrispondente relazione di ricorrenza.  
**Esempio:** Per l'array <a, b, c, r> la risposta sarà 0.  
Per l'array <b, a, r, c, a, r> la risposta sarà 2.

3. 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:



4. Un piroscafo collega regolarmente  $n$  città portuali. Ad ogni porto  $u$  il proprietario del piroscafo ricava  $P(u)$  euro, ma per andare dal porto  $u$  al porto  $v$  spende in totale  $C(u,v)$  euro. Se  $P(u)/C(u,v) > 1$ , il proprietario del piroscafo ha quindi realizzato un guadagno. Si vuole stabilire se esiste un itinerario ciclico di  $k$  città  $(x_0, x_1, \dots, x_k = x_0)$  per il quale

$$\frac{\sum_{i=1}^k P(x_i)}{\sum_{i=1}^k C(x_{i-1}, x_i)} > \mu$$

dove  $\mu \geq 1$  è una quantità stabilita a priori che rappresenta il guadagno minimo che il proprietario del piroscafo intende realizzare.

Si scriva un algoritmo che accetti in ingresso i ricavi  $P$  per ogni porto, i costi  $C$  per ogni coppia di porti e la costante  $\mu$ , e stabilisca se un tale itinerario esiste o meno. Si discuta la correttezza dell'algoritmo proposto e la sua complessità computazionale.