

Algoritmi e Strutture Dati

a.a. 2015/16

Compito del 17/05/2016

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:

31, 14, 34, 47

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 confrontino le seguenti funzioni utilizzando le relazioni O , Ω e Θ (giustificando le risposte):

$f(n)$	$g(n)$
$5n^3 + n^2 \log n + 3n^2$	$n^2 + \log (n^n \cdot n!)$
$n \log n$	$\log n!$
$n\sqrt{n}$	$n^2 + 3n$

3. Il problema di determinare i cammini minimi tra tutte le coppie di vertici di un grafo pesato $G=(V, E, w)$ con pesi positivi si può risolvere iterando $|V|$ volte l'algoritmo di Dijkstra o quello di Bellman-Ford. Si completi la tabella sottostante, specificando le complessità degli algoritmi indicati in funzione della tipologia di grafo utilizzato:

	Grafo sparso	Grafo denso
Iterated_Dijkstra (array)		
Iterated_Dijkstra (heap)		
Iterated_Bellman-Ford		

Supponendo che il grafo contenga pesi negativi, quale algoritmo conviene usare? Perché?

Algoritmi e Strutture Dati

a.a. 2015/16

Compito del 17/05/2016

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

Parte II

(2.5 ore; ogni esercizio vale 6 punti)

1. Dato un albero binario T
 - a. progettare una funzione **efficiente** che restituisca una copia T' di T , che contenga anche, in ogni nodo, la differenza fra l'altezza del suo sottoalbero sinistro e quella del suo sottoalbero destro.
 - b. Fornire la chiamata della funzione dal programma principale.
 - c. Modificare in modo adeguato il tipo dell'albero T per ottenere il tipo dell'albero T' .
 - d. Analizzare la complessità di tale funzione.

Il tipo **Node** utilizzato per rappresentare l'albero binario T 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** che, dato un vettore v di n interi **distinti** ($n \geq 2$), determina una coppia di elementi **distinti** x e y in v che minimizza $|x - y|$.

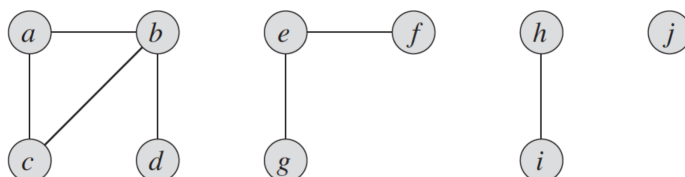
Discutere la complessità della soluzione proposta.

3. Il seguente algoritmo accetta in ingresso un grafo non orientato $G = (V, E)$.

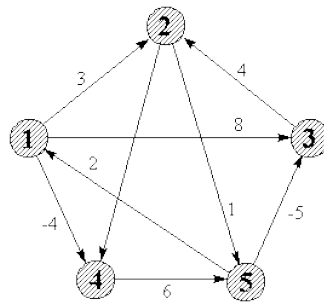
```
MyAlgorithm(G)  
1. n = 0  
2. for each vertex u ∈ G[V] do  
3.     MAKE-SET(u)  
4.     n = n + 1  
5. for each edge (u,v) ∈ G[E] do  
6.     if FIND-SET(u) ≠ FIND-SET(v) then  
7.         UNION(u,v)  
8.         n = n - 1  
9. return n
```

Cosa rappresenta il valore che **MyAlgorithm** restituisce al chiamante? Perché? (Giustificare formalmente la risposta.)

Si simuli inoltre accuratamente l'esecuzione dell'algoritmo sul grafo seguente, mostrando l'evoluzione delle strutture dati coinvolte passo dopo passo:



4. Si scriva l'algoritmo di Floyd-Warshall per determinare i cammini minimi tra tutte le coppie di vertici in un grafo, si derivi la sua complessità, e si simuli la sua esecuzione sul seguente grafo:



Si stabilisca infine se le seguenti affermazioni sono vere o false (fornendo una dimostrazione nel primo caso e un controesempio nel secondo):

- “Se la diagonale principale della matrice restituita in uscita dall'algoritmo di Floyd-Warshall contiene un elemento negativo, allora il grafo dato in ingresso contiene un ciclo negativo”.
- “Se la diagonale principale della matrice restituita in uscita dall'algoritmo di Floyd-Warshall contiene un elemento positivo, allora il grafo dato in ingresso contiene un ciclo positivo”.