

ESAME DI ALGORITMI E STRUTTURE DATI

A.A. 2015/2016 – APPELLO DEL 20/07/2016

TEMPO A DISPOSIZIONE: 120 MINUTI

ESERCIZIO 1:

Si consideri il seguente array:

1	2	3	4	5	6	7	8	9	10	11
42	20	38	16	15	28	32	12	10	7	6

- Si dica se tale array rappresenta o meno un MAX-HEAP;
- in caso affermativo, si mostri l'albero binario corrispondente.

ESERCIZIO 2:

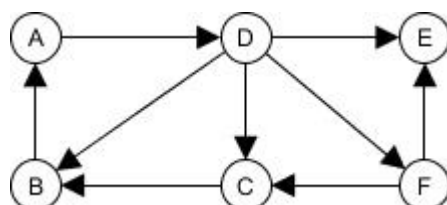
- Dare la definizione di Binary Search Tree.
- Si scriva lo pseudocodice dell'algoritmo di ricerca su BST

`search(chiave k) → elem`

- Indicare il costo dell'algoritmo di ricerca su BST `search(chiave k)` nel caso peggiore.
- Dire, motivando la risposta, se il costo nel caso peggiore dell'algoritmo di ricerca è diverso se l'albero considerato è un albero AVL.

ESERCIZIO 3:

Fornire la rappresentazione con *matrice di incidenza* del grafo orientato rappresentato in figura.



ESERCIZIO 4:

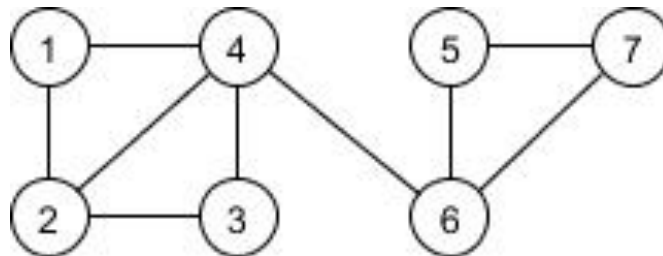
- a) Si consideri una **CODA** rappresentata come una tripla **(L, h, t)** dove:
- **L** è una LISTA semplice in cui ogni record è una coppia $(elem, next)$, dove $next$ è il puntatore al successivo record della lista;
 - **h** è il puntatore al primo record della lista;
 - **t** è il puntatore all'ultimo record della lista.

Scrivere lo pseudocodice che implementa le seguenti quattro operazioni:

- isEmpty() → boolean
- enqueue(elem e)
- dequeue() → elem e
- first() → elem e

ESERCIZIO 4:

- i) Si scriva lo pseudocodice dell'algoritmo `visitaDFS(vertex v)` → albero che esegue la visita in profondità di un grafo non orientato.
- ii) Descrivere i passi dell'algoritmo di visita quando applicato al grafo orientato rappresentato in figura partendo dal nodo **1** e presentare l'albero DFS prodotto dalla visita.



ESERCIZIO 6 (PER LA LODE):

Nel caso in cui non si possano evitare le collisioni in una tabella di hash, è necessario escogitare un metodo per risolverle. Indicare e descrivere due metodi per la risoluzione delle collisioni nelle tabelle di hash.