

# Esami di Stato per l'abilitazione all'esercizio della professione di Ingegnere

I Sessione anno 2005

Sezione B - Settore Informazione

**Classe 9 Ingegneria Informatica**

Tema 1

Prova pratica del 28 settembre 2005

Sviluppare un programma per rappresentare un grafo. Le funzioni richieste sono:

- `inserisciNodo (<nome>, <peso>)`  
riceve e inserisce le informazioni inerenti a un nodo. Emette una eccezione `NodoPresente` nel caso il nodo sia già presente nel sistema.
- `rimuoviNodo (<nome>)`  
rimuove dal sistema tutte le informazioni relative al nodo di nome `<nome>`. Emette una eccezione `NodoNonPresente` nel caso il nodo non sia presente nel sistema.
- `inserisciArco (<nome1>, <nome2>, <lunghezza>)`  
riceve i nomi di due nodi e una lunghezza e inserisce nel sistema l'informazione che fra i due nodi esiste un arco della lunghezza indicata. Se esiste già un arco tra i due nodi viene mantenuto solo l'arco di lunghezza minore.
- `rimuoviArco (<nome1>, <nome2>)`  
riceve i nomi di due piazze e rimuove dal sistema l'arco. Emette una eccezione `ArcoNonPresente` nel caso l'arco non esista.
- `trovaNodoPiuPesante ()`  
restituisce una `Collection` con il nome del nodo di peso maggiore (o i nomi dei nodi nel caso ci siano più nodi di uguale peso).
- `trovaNodoPiuConnessa ()`  
restituisce una `Collection` con il nome del nodo (o i nomi dei nodi) in cui converge il maggior numero di archi.
- `trovaNodiPiuDistanti ()`  
restituisce una `Collection` con il nome dei nodi in cui converge l'arco avente la maggiore lunghezza.
- `trovaPercorso (<nome1>, <nome2>)`  
riceve in ingresso il nome di due nodi e restituisce il percorso con la minima lunghezza totale che li connette. Emette una eccezione `PercorsoNonPresente` nel caso nessun percorso le connetta. Il percorso è reso come `LinkedList` di nomi di nodi.