Algoritmi I (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 17 Giugno 2008

Gli studenti esonerati devono svolgere solo gli esercizi 3 e 4. Nel caso in cui abbiano già consegnato la seconda parte del compito in un appello precedente (non ottenendo la sufficienza), il primo esonero non è più valido e l'esame deve essere sostenuto per intero.

Esercizio 1. Considerare un min-heap (heap in cui il minimo è nella radice) contenente n valori interi distinti, con $n \geq 15$, e rispondere alle seguenti domande:

- In quali nodi può venirsi a trovare il terzo elemento più piccolo? Disegnare un heap di dimensione 15 ed evidenziare i nodi legali e non, motivando la risposta: se u è un nodo legale, mostrare un'istanza in cui il terzo elemento più piccolo si trova in u; se u è un nodo non legale, dimostrare formalmente perché non lo è.
- Sia v il nodo contenente il terzo elemento più piccolo e sia k un intero positivo. Discutere come implementare, nel modo più efficiente possibile, un'operazione `DecrementaTerzo(v,k)` che diminuisce di k il valore del terzo elemento più piccolo. Analizzare nel caso peggiore il tempo di esecuzione dell'implementazione proposta.

Esercizio 2. Considerare il seguente pseudocodice:

```
    algoritmo analizzami(array A, indici i e j)
1.    if ( i < j ) then
2.        temp ← A[i]
3.        A[i] ← A[j]
4.        A[j] ← temp
5.        analizzami(A, i + 1, j − 1)
```

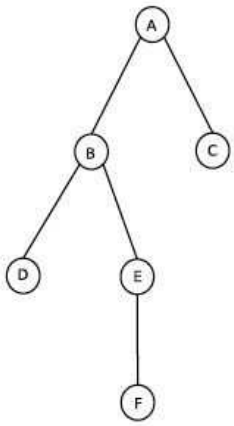
Che problema risolve l'algoritmo `analizzami`? Qual è il tempo di esecuzione nel caso peggiore? E nel caso migliore? Motivare le risposte.

Esercizio 3. Dimostrare o confutare con un controesempio le seguenti affermazioni:

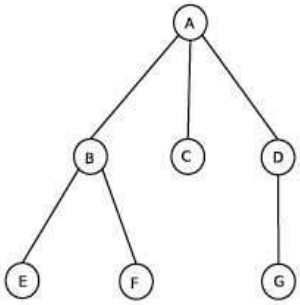
1. Dato un albero AVL T, è sempre possibile costruire T tramite una sequenza di operazioni `insert` tale che nessuna `insert` comporta una rotazione.
2. L'ordine in cui l'insieme di chiavi contenute in un albero binario di ricerca viene inserito è irrilevante: anche considerando ordini diversi, si ottiene comunque lo stesso albero.
3. Stessa affermazione del punto precedente, sostituendo "albero binario di ricerca" con "albero AVL".
4. Supporre di cancellare da un albero binario di ricerca due chiavi k_1 e k_2 contenute nei nodi u_1 ed u_2. Supporre inoltre che u_1 abbia due figli e che u_2 sia il predecessore di u_1. Cancellando k_1 prima di k_2 si ottiene lo stesso albero che si otterrebbe cancellando k_2 prima di k_1.

Esercizio 4. Un *matching perfetto* per un grafo non orientato $G = (V, E)$ è un insieme di archi $E' \subseteq E$ tale che ciascun nodo del grafo è incidente esattamente ad un arco di E'. Considereremo il problema del calcolo di un matching perfetto nel caso in cui il grafo sia un albero.

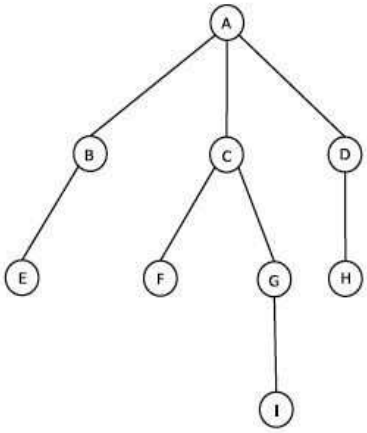
1. Per ciascuno dei seguenti alberi, mostrare quali archi fanno parte di un matching perfetto, oppure spiegare perché un matching perfetto non può esistere:



(i)



(ii)



(iii)

2. Progettare un algoritmo che, dato in input un albero, restitusce `true` se esiste un matching perfetto, e `false` altrimenti. Specificare se l'algoritmo assume che l'albero sia radicato e quali strutture dati utilizza (anche per la rappresentazione dell'albero). Discutere il tempo di esecuzione dell'algoritmo proposto, che dovrebbe essere lineare nel numero di nodi dell'albero.

Algoritmi I (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 17 Giugno 2008

Nome	Cognome

Soluzione Esercizio 1:

Algoritmi I (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 17 Giugno 2008

Algoritmi I (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 17 Giugno 2008

Nome	Cognome

Soluzione Esercizio 2:

Algoritmi I (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 17 Giugno 2008

Algoritmi I (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 17 Giugno 2008

Nome	Cognome

Soluzione Esercizio 3:

Algoritmi I (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 17 Giugno 2008

Algoritmi I (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 17 Giugno 2008

Nome	Cognome

Soluzione Esercizio 4:

Algoritmi I (A.A. 2007-2008) Docente: Irene Finocchi

Appello del 17 Giugno 2008