

**“ALGORITMI 3”**  
**CORSO DI STUDIO IN INFORMATICA (laurea specialistica)**  
**UNIVERSITÀ DEGLI STUDI DI CATANIA**  
**ANNO ACCADEMICO 2006/07**

2<sup>a</sup> prova in itinere – 30 Maggio 2007

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

**ESERCIZIO 1**

Si eseguano nell'ordine dato le seguenti operazioni su un heap di Fibonacci inizialmente vuoto:

- INSERT 27, 17, 19, 20, 24, 12, 11, 10, 14, 18
- EXTRACT-MIN
- DECREASE-KEY(19,7)<sup>1</sup>
- DELETE(17)
- DECREASE-KEY(24,5)<sup>1</sup>
- EXTRACT-MIN

Dopo ciascuna operazione, si disegni l'heap di Fibonacci risultante, tenendo conto che nella lista delle radici i nuovi elementi vanno inseriti a destra del minimo corrente e che il consolidamento della lista delle radici ha sempre inizio nel nodo a destra del minimo appena cancellato.

**ESERCIZIO 2**

(a) Si consideri la seguente procedura per incrementare il valore di una chiave in un heap binomiale:

```
MY-INCREASE-KEY( $x, k$ )
  Se  $KEY[x] \geq k$ , ERRORE.
  Se  $x$  è una foglia, si ponga  $KEY[x] := k$ 
  altrimenti
    - si determini il figlio  $y$  di  $x$  avente chiave minima
    - si ponga  $KEY[x] := KEY[y]$ 
    - si chiami ricorsivamente MY-INCREASE-KEY( $y, k$ )
```

Si determini la complessità della procedura MY-INCREASE-KEY nel caso pessimo e se ne dimostri la correttezza nel caso degli heap binomiali.

Si proponga un algoritmo più efficiente della procedura MY-INCREASE-KEY per incrementare il valore di una chiave in un heap binomiale indicandone la complessità nel caso pessimo.

(b) Si eseguano le operazioni indicate nell'Esercizio 1 su un heap binomiale inizialmente vuoto.

---

<sup>1</sup>Per semplicità chiavi e nodi sono stati identificati.