

Esame di Laboratorio di Algoritmi

Università degli Studi di Catania - Corso di Laurea in Informatica

Testo della Prova

1. Si fornisca una classe C++, denominata `MaxHeap<H>`, che implementi la seguente interfaccia `BHeap<H>`, che rappresenta uno Heap Binario e contenente i seguenti metodi virtuali.
 - (a) `Heap<H>* enqueue(H x)` inserisce un nuovo elemento nella struttura dati e restituisce un puntatore ad un oggetto di tipo `BHeap<H>`. La procedura non ha effetto se la struttura dati risulta piena al momento dell'inserimento;
 - (b) `H* top()` restituisce il puntatore all'elemento che si trova in testa alla struttura struttura, oppure NULL se la struttura è vuota;
 - (c) `BHeap<H>* extract()` è la procedura estrae l'elemento che si trova in testa alla struttura e restituisce un puntatore ad un oggetto di tipo `BHeap<H>`. La procedura restituisce NULL se la struttura dati risulta vuota al momento dell'estrazione;
 - (d) `void buildHeap(H *v, int n)` è la procedura che, preso in input un vettore `v` contenente `n` elementi di tipo `H`, costruisce uno Heap binario. La procedura restituisce un puntatore ad un oggetto di tipo `BHeap<H>`.
 - (e) `void print()` è la procedura che stampa gli elementi dello Heap dal primo all'ultimo livello e, per ciascun livello, da sinistra verso destra;

Si crei quindi un'istanza di `MaxHeap<int>` a partire dal vettore di interi contenente i 9 seguenti elementi:

[5, 3, 2, 8, 6, 1, 13, 4, 14]

in seguito si inseriscano al suo interno, mediante la procedura `enqueue()`, i seguenti valori:

7,10,12

Si eseguano in seguito due operazioni `extractMax()`,

Si esegua in seguito la stampa dei valori inseriti nella struttura.

L'output del programma sarà quindi:

12, 10, 7, 5, 8, 1, 2, 4, 3, 6

```
template <class H> class BHeap {
public:
    virtual BHeap<H>* enqueue(H x) = 0;
    virtual H* top() = 0;
    virtual BHeap<H>* extract() = 0;
    virtual BHeap<H>* buildHeap() = 0;
    virtual void print() = 0;
}
```

•••