

Lista #5

Módulo 2 - Hash, Heap e Partição Dinâmica

Data de entrega: 14 de maio de 2017

Acompanham esta lista os arquivos **mod2_lista5.h** e **mod2_lista5.cpp**, contendo as declarações iniciais. Modifique estes arquivos para implementar as questões pedidas e envie-os de volta zipados com nome no padrão <numero_matricula>.zip por email para **profs-eda@tecgraf.puc-rio.br**, com o assunto **[EDA] Lista 5**. Atenção: Crie um arquivo contendo a função main do seu programa para testar suas implementações, mas envie SOMENTE os arquivos e as classes solicitadas.

1. (Declaração da classe no final do documento) Implemente uma classe Heap que contém chaves inteiras, armazenadas em um vetor, de forma que, para cada nó de índice i , tem-se:

- índice do filho esquerdo de i : $2*i + 1$
- índice do filho direito de i : $2*i + 2$
- índice do pai de i : $(i-1)/2$

O heap implementado será do tipo MIN-HEAP, ou seja, cada nó é **menor** que os seus filhos.

Um dos construtores recebe um vetor de inteiros em qualquer ordem, e deve construir propriamente o heap a partir deles, em $O(n)$.

As demais funções que alterem o heap, inserindo ou removendo elementos, devem garantir que todas as propriedades do heap continuem mantidas.

A função **show** exibe os elementos do heap na ordem armazenada no vetor, separados por espaço, e já está implementada. Por favor, não a modifique.

2. Faça as seguintes alterações na sua classe Heap implementada na questão 1:
 - (a) Passe a utilizar um `std::vector<int>` para armazenar os elementos. Observe que `MAX_ELEMENTS` e `_size` não serão mais necessários. Faça outras modificações necessárias em todo o código, para fazer essa adaptação.
O construtor que recebe os elementos também deve ser modificado para:

```
Heap(const std::vector<int>& values);
```

- (b) Transforme a classe Heap em uma classe template, de forma que o usuário da classe possa definir o tipo das chaves que deseja utilizar no momento em que a instancia. Por exemplo:

```
Heap<float> myHeap;  
myHeap.push(0.5f);  
(...)
```

Novamente, faça todas as modificações necessárias para essa adaptação. E faça testes com os tipos básicos int, float e double.

Segue a declaração inicial da classe Heap, para a questão 1.

Envie somente a versão modificada (com template e std::vector).

```
#ifndef MOD2_LISTA5_H  
#define MOD2_LISTA5_H  
  
#include <string>  
  
#define MAX_ELEMENTS 50  
  
class Heap /* Min-Heap */  
{  
public:  
    // Construtor vazio  
    Heap();  
  
    // Constrói o heap a partir de um vetor de elementos  
    Heap(int* values, unsigned int size);  
  
    // Constrói o heap a partir de outro heap  
    Heap(const Heap& other);  
  
    // Destroi o heap  
    ~Heap();  
  
    // Retorna o numero de elementos atualmente no heap  
    unsigned int size();  
  
    // Insere um novo elemento  
    void push(int value);  
  
    // Retorna o elemento do topo  
    int top();
```

```

    //Remove o elemento do topo
    void pop();

    //Nao modificar:
    //Exibe os elementos do heap
    void show(const std::string& title);

private:

    /* Funcoes privadas que julgar necessarias */

private:
    //Contem os elementos do heap
    int _container[MAX_ELEMENTS];

    //Tamanho atual do heap
    unsigned int _size;
};

#endif

```