DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Estruturas de Dados

PA03 - Heap binário

Professores: Wagner Meira Jr
Eder Figueiredo

Roteiro da atividade

Nesta atividade, você deve implementar um programa que manipula uma estrutura de Min Heap. O programa deve:

1. Lê da entrada padrão:
   - Um número inteiro n, indicando o tamanho do **Min Heap**.
   - **n** inteiros, representando os elementos a serem inseridos no Min Heap.
2. Inserir os **n** elementos na estrutura de dados na ordem de leitura.
3. **Remover** todos os elementos do Min Heap e imprimi-los na ordem de remoção, separados por um espaço em branco. Finalize a saída com uma quebra de linha.

Exemplo

Entrada:

```
7

13 15 8 23 21 9 2
```

Saída esperada:

```
2 8 9 13 15 21 23
```

Exemplo

Entrada:

```
5
```

```
5 4 3 2 1
```

Saída esperada:

```
1 2 3 4 5
```

Exemplo

Entrada:

```
3

1 2 3
```

Saída esperada:

```
1 2 3
```

Requisitos de Implementação

- **Em C:**
  - Implementar o TAD Heap conforme o arquivo Heap.h
- **Em C++:**
  - Implementar a classe Heap conforme o arquivo Heap.hpp
- Restrições:
  - Não modificar a interface das funções/métodos.
  - Não adicionar ou remover campos e métodos da estrutura ou classe.
  - É permitido adicionar #include para bibliotecas necessárias.

Heap.h - Interface do TAD Heap e suas funções

```
typedef struct s_heap{
    int tamanho;
    int* dados;
} Heap;
Heap* NovoHeap(int maxsize);
void DeletaHeap(Heap* h);
void Inserir(Heap* h, int x);
int Remover(Heap* h);
int GetAncestral(Heap* h, int posicao);
int GetSucessorEsq(Heap* h, int posicao);
int GetSucessorDir(Heap* h, int posicao);
//Retorna 1 caso h esteja vazio, 0 caso contrário.
int Vazio(Heap* h);
//Funções necessárias para implementar o Heapify recursivo
void HeapifyPorBaixo(Heap* h, int posicao);
void HeapifyPorCima(Heap* h, int posicao);
```

Heap.hpp - Interface da classe Heap

```
class Heap{
    public:
        Heap(int maxsize);
        ~Heap();
        void Inserir(int x);
        int Remover();
        //Retorna 1 caso h esteja vazio, 0 caso contrário.
        bool Vazio();
    private:
        int GetAncestral(int posicao);
        int GetSucessorEsq(int posicao);
        int GetSucessorDir(int posicao);
        int tamanho;
        int* data;
       //Funções necessárias para implementar o Heapify recursivo
        void HeapifyPorBaixo(int posicao);
        void HeapifyPorCima(int posicao);
};
```

Submissão

- A submissão será feita por **VPL**. Certifique-se de seguir as instruções do tutorial disponibilizado no moodle.
- O seu arquivo executável **DEVE** se chamar **pa3.out** e deve estar localizado na pasta **bin**.
- Seu código será compilado com o comando:

make all

- Você **DEVE** utilizar a estrutura de projeto abaixo junto ao Makefile :

```
- PA3
    |- src
    |- bin
    |- obj
    |- include
    Makefile
```