

Estruturas de Dados

PA02 - Heap binário

Professores: Wagner Meira Jr
Eder Figueiredo

Roteiro da atividade

Nesta atividade você deverá implementar um programa que:

1. Lê da entrada padrão:
 - ☐ Um inteiro n dizendo qual será o tamanho do **Min Heap**.
 - ☐ Seguido de n inteiros dizendo quais são os elementos a serem inseridos no **Min Heap**.
2. Você deve inserir os n elementos na ordem de leitura no **Min Heap**.
3. Após inserir você deve remover todos os elementos do **Min Heap** e imprimi-los na tela, em ordem de remoção e separados por um espaço em branco. Encerre sua impressão 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

Roteiro da atividade

Observações sobre a implementação:

- **Em C:** Você deve implementar o TAD Heap bem como as funções propostas no arquivo Heap.h
- **Em C++:** Você deve implementar a classe do arquivo Heap.hpp.
- Em ambos os casos você não deve alterar a interface das funções ou métodos(você não pode adicionar campos ou métodos a classe, nem modificar as assinaturas dos métodos ou funções), mas pode adicionar os includes de seus TADs.

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 **pa2.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 :
 - PA2
 - |- src
 - |- bin
 - |- obj
 - |- include
- Makefile