

Lista 8 – Fila de Prioridade e Heapsort

ESTRUTURA DE DADOS I – Pedro Nuno Moura

Monitor: Celio Ferreira Camara Junior

Para as questões abaixo, considere as implementações de Heap Máxima e Heap Mínima vistas em sala de aula.

ATENÇÃO: Para todas as questões, deve ser informada e explicada a complexidade computacional alcançada.

1) Defina o conceito de Heap e dê um exemplo de uma Heap.

2) Para a classe `HeapBinariaMinima`, crie um construtor que receba uma Heap Máxima como parâmetro e a converta em Heap Mínima. Esse construtor deve possuir o protótipo abaixo:

```
public HeapBinariaMinima(HeapBinariaMaxima heapMaxima);
```

3) Implemente agora um construtor que faça uma operação análoga para a classe `HeapBinariaMaxima`. Esse construtor deve seguir o seguinte protótipo:

```
public HeapBinariaMaxima(HeapBinariaMinima heapMinima);
```

4) Desenvolva um método que receba uma Árvore Binária Completa como um objeto da classe `Arvbin<T>` e a modifique de modo a transformá-la em uma Heap Máxima. O método deve seguir o protótipo abaixo:

```
public void transformaHeapMaxima(Arvbin<T> arvore);
```

5) Implemente um método não recursivo que verifica se uma árvore armazenada em um vetor de inteiros corresponde a uma Heap Máxima, isto é, se todo nó possui uma chave maior ou igual às chaves dos seus filhos. O método deve possuir o seguinte protótipo:

```
public boolean verificaPropriedadeHeap(int[] vetor);
```

6) Implemente um método que remova e retorne o elemento na i -ésima posição do vetor representativo de uma Heap Máxima. O protótipo a ser seguido está abaixo:

```
public boolean remove(int i);
```

7) Mostre passo a passo como o seguinte vetor de inteiros é ordenado pelo método *Heapsort*. Você deve mostrar apenas a sequência de alterações que são feitas no vetor a cada passo do algoritmo.

22 4 35 12 3 18 12 55