

PCO001 – Algoritmos e Estruturas de Dados

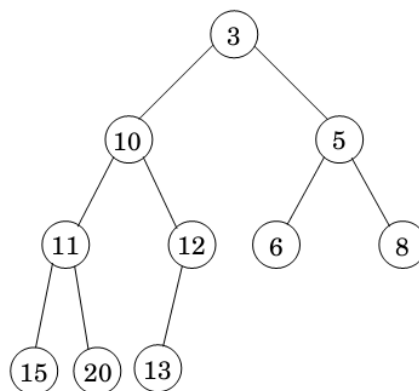
UNIFEI - Universidade Federal de Itajubá

Prof. João Paulo R. R. Leite (joaopaulo@unifei.edu.br)

Formato: Arquivo CPP ÚNICO contendo o código requisitado abaixo. **Caso o arquivo não esteja no formato pedido, será desconsiderado.**

Avaliação: O relatório valerá 3 pontos na N2 da disciplina de PCO001, que correspondem a 30%.

Escreva um programa em C++ que implemente uma estrutura de “**Heap de Mínimo**”. Nessa estrutura, o **menor elemento está sempre no topo**, e cada elemento é sempre menor do que todos os elementos que estão abaixo dele na *heap*. Veja um exemplo:



A heap deverá ser representada como um **vetor de inteiros** com capacidade máxima para 15 elementos e um atributo responsável por manter a **quantidade atual** de elementos na heap.

No programa, você terá que escrever as seguintes funções:

- Inserir um novo elemento na *heap*, chamada **push** (void). Se o usuário tentar inserir um elemento em uma heap cheia, mostre uma mensagem de advertência (“Heap cheia.”).
- Remover um elemento da *heap*, chamada **pop** (void). Sempre remove o elemento do topo. Se o usuário tentar remover um elemento em uma heap vazia, mostre uma mensagem de advertência (“Heap vazia.”).
- Retornar o elemento no topo da *heap*, sem removê-lo, chamada **top** (int).
- Verificar se a *heap* está vazia, chamada **empty** (bool).
- Verificar se a *heap* está cheia, chamada **full** (bool).
- Imprimir a heap em formato de vetor (linear), chamada **print** (void). Após cada operação de push/pop, imprimir o estado da heap.
- Uma função de ajuste (**heapfy**), que serve para ajustar a *heap* todas as vezes que o usuário insere ou remove um elemento. Lembre-se de que a heap tem sempre que manter sua estrutura de mínimo, como mostrado na figura acima. Nós vimos uma função semelhante nas aulas, mas era uma *heap* de máximo.