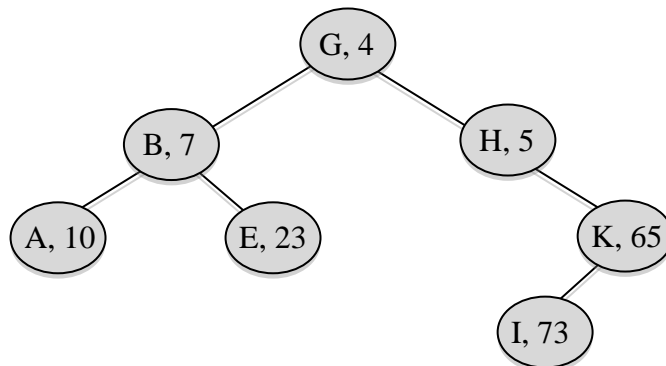


CAP234 - Computação Aplicada I
Lista de Exercícios 06 - Prof. Senne
Data-limite de entrega: 14/05/2010

1. Implemente a função **HeapMin(v, n, i)** para a construção de *heap* mínimo a partir do vetor v de n elementos não ordenados, onde i é o índice do próximo nó a ser incluído no *heap*.
2. Considere que uma fila de prioridades utiliza a estrutura de dados de um *heap* mínimo. Implemente as operações **Minimo**, **ExtrairMinimo**, **DiminuirChave** e **Inserir** sobre esta fila de prioridades.
3. Um **treap** (mistura de "*tree*" com "*heap*") é uma árvore binária de procura em que cada nó possui uma chave e um valor de prioridade. Considere que todas as chaves são distintas e todos os valores de prioridade são distintos. Os nós de um *treap* são ordenados de forma que as chaves obedecem à propriedade de árvore binária de procura e que as prioridades obedecem à ordem de *heap* mínimo, ou seja:
 - Se v é um filho à esquerda de u , então $\text{chave}(v) < \text{chave}(u)$;
 - Se v é um filho à direita de u , então $\text{chave}(v) > \text{chave}(u)$;
 - Se v é um filho de u , então $\text{prioridade}(v) > \text{prioridade}(u)$.

A figura a seguir mostra um exemplo de *treap*, em que as chaves são letras e as prioridades são valores inteiros.



Para implementar a inclusão de nós em um *treap* pode-se executar o procedimento habitual de inclusão em árvore binária de procura e, em seguida, executar rotações para restaurar a ordem de *heap* mínimo. Implemente a operação **Incluir** em um *treap*.