DCA0214.1 - LABORATÓRIO DE ESTRUTURAS DE DADOS
Aula 10: Listas de prioridades e Heaps

Prof. Felipe Fernandes

04 de Outubro de 2019

1. Escreva uma função que recebe um um vetor $V[1...n]$ e decide se V é um *heap max* ou min ou nenhuma dessas possibilidades.

2. Implemente as seguintes operações básicas para *heap max*:

   (a) Subir (iterativo e recursivo)

   (b) Descer (iterativo e recursivo)

   (c) Construção

   (d) Remoção

3. Utilizando uma *heap max*, implemente o algoritmo de ordenação *HeapSort*.

4. Dados um vetor $V[1...n]$ de inteiros positivos **desordenados** e dois inteiros positivos x e k, implemente um procedimento que retorna os k valores em V mais próximo de x. Exemplo: $V = \{10, 2, 14, 4, 7, 6\}, x = 5, k = 3$. Os três valores mais próximos de 5 são: 4, 6 e 7. Sua implementação deve ter complexidade $O(nlogk)$.

5. Um ladrão invadiu um cofre de banco, onde $n \geq 1$ barras de ouro estão guardadas. As barras de ouro estão numeradas de 1 a n. Cada barra $i \in \{1, ..., n\}$ possui uma massa, em kg, denotada por w_i, e possui também um valor equivalente em reais, denotado por v_i. O ladrão possui uma mochila, na qual ele pretende transportar as barras de ouro roubadas. A mochila do ladrão pode transportar uma carga (massa) máxima de W quilogramas. Suponha que o ladrão possui um mecanismo de quebrar uma barra de ouro a fim de, caso necessário, fracioná-la e facilitar seu transporte. Massa e valor são igualmente distribuídos por cada fração de uma barra. Por exemplo, se uma barra i tem $v_i = R\$120$ e $w_i = 30kg$, então $\frac{2}{3}$ da barra i valem $v_i = R\$80$ e $w_i = 20kg$. Sua tarefa é ajudar o ladrão a escolher um subconjunto de barras de ouro a serem roubadas de modo a maximizar o valor em reais obtido na ação, respeitando o peso carregado na mochila do meliante. Considere que o ladrão pode entrar no cofre do banco apenas

uma vez (ou seja, ele não pode regressar). Implemente um algoritmo que recebe w_i, v_i, $i \in 1, ..., n$, e W e retorna o valor (máximo) do roubo. Complexidade requerida no pior caso: $O(nlogn)$. OBS.: você não pode utilizar nenhum algoritmo de ordenação.