Análise de algoritmos - Lista 3

Professor: Guilherme Oliveira Mota

Entrega: 23h55 de 30/7/2018 (**Somente pelo Tidia**)

(1) O problema da Mochila Inteira é semelhante ao da mochila fracionária, mas não é permitido adicionar à mochila somente frações dos itens. A estratégia gulosa de escolher sempre o item com maior *valor/peso* encontra uma solução ótima para o problema da Mochila Inteira? Prove ou mostre um contra-exemplo.

(2) Execute o algoritmo de Huffman passo a passo na entrada $A = \{a, b, c, d, e, g\}$ onde $f_a = 3$, $f_b = 2$, $f_c = 6$, $f_d = 8$, $f_e = 2$ e $f_g = 6$.

(3) Mostre que se todos os pesos das arestas são distintos, então o grafo tem somente uma árvore geradora mínima. *Dica:* por contradição, suponha que o grafo tem duas árvores geradoras mínimas diferentes e utilize um *argumento de troca.*

(4) Escreva um algoritmo que recebe uma árvore construída pelo algoritmo de Huffman, um alfabeto, uma sequência de bits e retorna a string decodificada.

(5) Quantas árvores geradoras mínimas possui o grafo da Figura 1? Execute passo a passo o algoritmo de Kruskal sobre ele para gerar uma delas.

(6) Para o problema de encontrar uma árvore geradora mínima em um grafo $G = (V, E)$, vimos em sala o algoritmo de Kruskal e um outro algoritmo bem conhecido é o algoritmo de Prim (veja notas de aula).

Um terceiro algoritmo guloso clássico para esse problema é o algoritmo de *Boruvka*, que funciona como segue: Cada vértice começa como uma componente conexa (similar ao Kruskal). Repita o seguinte procedimento até tenhamos somente uma componente conexa: para cada componente C, adicione a aresta de menor peso que conecta C a uma outra componente (isso gera componentes maiores).

- Escreva o pseudocódigo para esse algoritmo, explicando o que cada linha faz;
- Mostre a execução desse algoritmo sobre o grafo da Figura 2;
- Explique (de forma sucinta) qual o tempo de execução do algoritmo.

(7) Considere um inteiro positivo n e m conjuntos $C_1, \ldots, C_m \subseteq \{1, \ldots, n\}$. Uma *cobertura* é um conjunto de índices $I \subseteq \{1, \ldots, m\}$ tal que

$$\bigcup_{i \in I} C_i = \{1, \ldots, n\}.$$

Uma *cobertura mínima* é uma cobertura com a menor quantidade possível de conjuntos. Dizemos que $\bigcup_{i \in I} C_i$ é o conjunto dos elementos não *cobertos* por I.

O problema da *Cobertura de conjuntos* consiste em, dados n e $C_1, \ldots, C_m \subseteq \{1, \ldots, n\}$, encontrar a cardinalidade de uma cobertura mínima. Considere o seguinte algoritmo guloso para o problema.

Algoritmo 1: COBERTURA CONJUNTOS(n, $C_1, \ldots, C_m \subseteq \{1, \ldots, n\}$)

```
I = ∅
enquanto I não é cobertura faça
    i é um índice tal que S_i contém a maior quantidade de elementos não cobertos por I
    I = I ∪ {i}
retorna I
```

Esse algoritmo nem sempre encontra uma cobertura mínima para o problema. Mostre uma instância, i.e., descreva conjuntos $C_1, \ldots, C_m \subseteq \{1, \ldots, n\}$ tal que a cobertura mínima tem somente 2 conjuntos, mas o algoritmo COBERTURA CONJUNTOS retorna uma cobertura I com $\Omega(\log n)$ conjuntos.

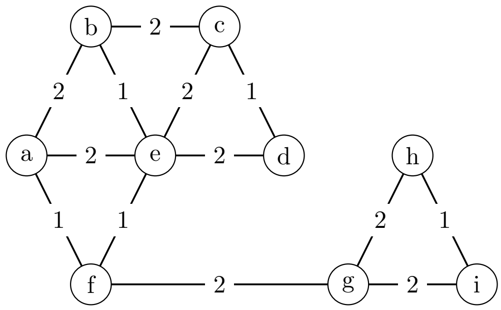

Figura 1. Grafo A.

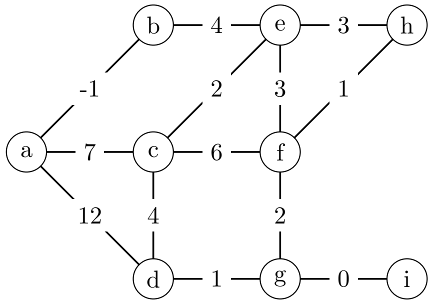

Figura 2. Grafo B.