

# CPS740 - Algoritmos e Grafos - 2020/2 - Lista 2

Programa de Engenharia de Sistemas e Computação – COPPE/UFRJ  
Engenharia de Computação e Informação - Escola Politécnica/UFRJ

Professores: Celina e Franklin

Monitores: Bruno, Edinelço, Guilherme, Luiz e Vitor

Data de entrega: 05/08/2020

**Questão 1** - No problema da mochila (*knapsack problem*) são dados um limite  $P$  e  $n$  objetos onde cada objeto tem peso  $p_i$  e valor  $v_i$ . Queremos determinar o subconjunto de objetos com soma de pesos menor ou igual a  $P$  tal que a soma dos valores dos objetos selecionados seja máxima.

(a) Proponha um algoritmo guloso para resolver esse problema. Indique um contra-exemplo para o qual o seu algoritmo não funciona e explique o porquê.

(b) Descreva um algoritmo para resolver esse problema utilizando a técnica da programação dinâmica. O seu algoritmo deve determinar quais objetos pertencem ao subconjunto viável de valor máximo.

**Questão 2** - Considere o seguinte problema. A partir de um conjunto de tarefas que possuem um tempo de início e um tempo de término, você quer realizar o maior número possível de tarefas, porém, as tarefas não podem se sobrepor, ou seja, duas ou mais tarefas que você fizer não podem ter um tempo em comum. Dada uma lista de tarefas, onde cada tarefa corresponde a um par contendo o tempo de início  $t_i$  e o tempo de término  $t_f$ , desenvolva um algoritmo que retorne o número máximo de tarefas que você pode realizar. Note que uma tarefa que termina no tempo  $t_j$  não é considerada sobreposta com uma tarefa que começa no tempo  $t_j$ . **Exemplo:** Para a lista de 4 tarefas  $T = \{(2, 5), (11, 15), (4, 9), (7, 10)\}$ , o número máximo é 3, correspondendo a realizar por exemplo as tarefas (2, 5), (11, 15) e (7, 10).

**Questão 3** - Considere o grafo  $G$  da Figura 1 e responda:

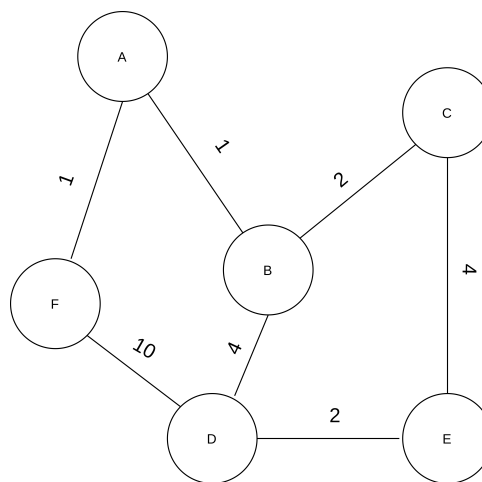


Figura 1: Grafo  $G$

(a) Quantas árvores geradoras mínimas tem o grafo  $G$ ? Desenhe ou indique as arestas que as compõem.

(b) Porque a aresta  $(F, D)$  não pode compor uma árvore geradora mínima para  $G$ ?

(c) Qual é o número cromático  $\chi(G)$ ?

**Questão 4 -** Considere as formas de representação de grafos: matriz de adjacências e estrutura de adjacências<sup>1</sup> e responda

(a) Considere que ambas as estruturas já estão prontas e carregadas na memória. Para ambas, descreva ou escreva um algoritmo que, primeiramente, checa se existe uma aresta entre os vértices  $u$  e  $v$  e, depois, deleta essa aresta, caso exista. Informe, também, a complexidade do seu algoritmo.

(b) O que mudaria no algoritmo proposto no item (a) se, para a estrutura de adjacências, no lugar de um vetor de listas encadeadas, a estrutura fosse um vetor de vetores, nos quais as arestas são ordenadas em ordem crescente? Veja a estrutura na Figura 2.

*Dica: Fale primeiro das diferenças de detecção da aresta e depois das de remoção.*

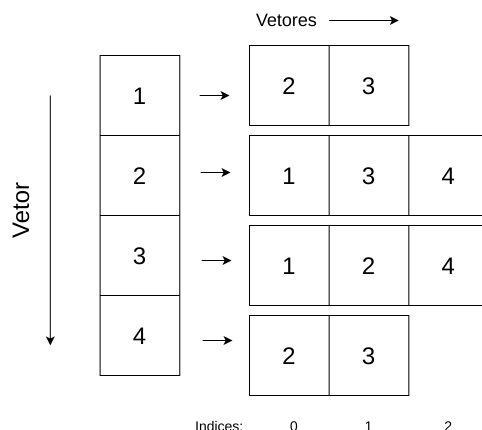


Figura 2: Representação de um grafo por um vetor de vetores ordenados.

**Questão 5 - O algoritmo de Kruskal.** Considere o seguinte algoritmo para grafos com pesos em arestas, assumindo que  $sort(E)$  é uma função que ordena o conjunto das arestas nos seus pesos em ordem crescente e que  $min(E)$  é uma função que retorna o elemento de menor peso de  $E$ .

---

**Algoritmo:** Mistério ( $G$ )

**Entrada:** Um grafo  $G = (V, E)$

- 1:  $M \leftarrow \emptyset$
  - 2:  $E \leftarrow sort(E)$
  - 3: **enquanto**  $M$  não for uma árvore geradora **faça**
  - 4:    $e \leftarrow min(E)$
  - 5:   Adicione  $e$  em  $M$
  - 6:   Remova  $e$  de  $E$
  - 7: **retorne**  $M$
- 

(a) Mostre e justifique um exemplo no qual o algoritmo Mistério não retorna uma árvore geradora mínima.

(b) Dê uma condição para adicionar ou não  $e$  em  $M$ , na linha 5, de modo que o algoritmo assim modificado sempre retorne uma árvore geradora mínima de  $G$ . Justifique.

*Dica: durante a execução do algoritmo, o que acontece com  $M$  caso se adicione uma aresta e que feche um ciclo em  $M$ ?*

(c) Denote por  $E(M)$  e  $V(G)$  as arestas de  $M$  e os vértices de  $G$  respectivamente. Denote por  $|X|$  o número de elementos de um conjunto  $X$ . Com a modificação do item anterior, mostre que alterando a linha 3 para “**enquanto**  $|E(M)| < |V(G)| - 1$  **faça**” não altera a corretude do algoritmo.

---

<sup>1</sup>Por exemplo, vetor de ponteiros para listas encadeadas.

**Questão 6** - O objetivo dessa questão é tentar contar o número de árvores geradoras de um grafo completo rotulado com  $n$  vértices, que será denotado por  $T_n$ .

(a) Primeiramente, para os valores pequenos de  $n$ , qual o valor de  $T_n$ ? Determine  $T_2$ ,  $T_3$  e  $T_4$ . Além disso, dê um palpite para  $T_n$  e justifique. Observe que  $T_n$  é equivalente a quantidade de árvores rotuladas com  $n$  vértices.

(b) Definimos uma **sequência de Prüfer**, denotada por  $S_n$ , como o número de seqüências de tamanho  $n - 2$  de um conjunto com  $n$  elementos. Fixada uma árvore rotulada  $T$  com  $n$  vértices, quantas seqüências de *Prüfer* existem para os vértices de  $T$ ?

(c) Denotando por  $\mathbb{S}$  e  $\mathbb{T}$  o conjunto de todas as seqüências de *Prüfer* de  $n$  vértices e o conjunto de todas as árvores rotuladas com  $n$  vértices, respectivamente, mostre que existe uma injeção de  $\mathbb{T}$  em  $\mathbb{S}$ . Ou seja, podemos codificar uma árvore rotulada com  $n$  vértices em uma seqüência de tamanho  $n - 2$  de maneira injetiva.

(d)(bônus) Mostre que existe uma injeção de  $\mathbb{S}$  em  $\mathbb{T}$ , implicando que conseguimos recuperar toda informação de uma árvore rotulada com  $n$  vértices, a partir de uma seqüência de tamanho  $n - 2$ . Conclua que  $S_n = T_n$ .