

# Construção e Análise de Algoritmos

## lista de exercícios 17

### 1. Arestas unitárias

O problema dos caminhos mais curtos fica mais fácil se todas as arestas do grafo possuem distância igual a 1?

Apresente um algoritmo para essa versão do problema dos caminhos mais curtos que executa em tempo menor que  $O(m \log n)$ .

(Dica: é possível resolver esse problema em tempo  $O(m)$ .)

### 2. Arestas 0-1

O problema dos caminhos mais curtos com arestas unitárias fica mais difícil se nós permitimos que algumas arestas possuam distância igual a 0?

Apresente um algoritmo tão rápido quanto o anterior para essa versão do problema.

### 3. Múltiplos caminhos mais curtos

É comum que existam vários caminhos com distância mínima entre dois vértices em um grafo direcionado.

Modifique o algoritmo apresentado na aula 17 para que ele passe a calcular também o número de caminhos mais curtos de um vértice  $v$  até todos os outros vértices do grafo.

### 4. Arestas com distância negativa

Em alguns contextos, pode fazer sentido trabalhar com distâncias negativas associadas às arestas do grafo.

a) Apresente um exemplo de grafo que contém arestas com distâncias negativas onde o nosso algoritmo guloso não encontra uma solução ótima para o problema dos caminhos mais curtos.

b) Suponha que alguém tenha a ideia somar um número positivo  $K$  muito grande às distâncias de todas as arestas do grafo (para que elas passem a ser todas positivas), depois executar o nosso algoritmo guloso, e depois subtrair o número  $K$  de todas as arestas da solução.

Essa ideia permite encontrar a solução ótima em um problema de caminhos mais curtos com distâncias negativas?

Explique.

## 5. Caminhos de tamanho limitado

Suponha que você deseja encontrar caminhos mais curtos de um vértice  $v$  a todos os outros vértices do grafo, que possuam no máximo  $k$  arestas.

- a) Apresente um algoritmo que resolve essa versão do problema dos caminhos mais curtos.
- b) Argumente que o seu algoritmo encontra uma solução ótima para o problema.
- c) Analise a complexidade do seu algoritmo.