

# Exercício de Programação 4: Diâmetro do grafo

11 de Maio de 2020

## 1 Introdução Teórica

Vimos na última tarefa que um simples algoritmo de busca em profundidade pode nos dar informações úteis sobre um grafo como sua conectividade e a presença de ciclos. Com isso pudemos calcular o número e o tamanho das componentes conexas do grafo do nosso grafo dos encontros. Uma busca em profundidade nos indica se há um caminho entre dois nós, mas não diz nada sobre o menor caminho. Para encontrar o menor caminho, precisamos usar um outro algoritmo, uma *busca em largura*.

Esse tema é tratado nas páginas 538 a 542 da quarta edição do livro Algorithms [SW11]. Na busca em profundidade começamos de um vértice exploramos o primeiro vizinho deste vértice que ainda não foi explorado. Implementamos isso recursivamente, mas seria equivalente a colocar os novos vizinhos em uma pilha e sempre seguir o nó que está no topo e é desempilhado. Na busca em largura colocamos os vizinhos não explorados em uma fila e seguimos o caminho pela ordem de chegada. Dessa forma vamos construindo o caminho mais curto entre o nó fonte e todos os demais nós.

No nosso exemplo, a distância entre os nós no grafo indicam quantos passos de contaminação precisam acontecer até uma pessoa ser infectada. Olhando para a componente gigante do grafo, temos duas métricas que podemos calcular que podem ser úteis. O *diâmetro* do grafo é a maior distância entre dois nós. Quanto maior o diâmetro mais passos de transmissão precisam ser dados para que todos se infectem. Uma métrica provavelmente mais útil é a *distância média* entre os vértices.

## 2 Tarefa

Para a quarta tarefa vamos desconsiderar os nós desconectados e vamos investigar apenas a componente gigante do grafo dos encontros. Usando a busca em largura, vocês devem calcular o histograma das distâncias entre pares de nós. Ou seja, vocês devem desenhar um gráfico que indique quantos pares de nós são adjacentes, quantos estão a 2 passos de distância, quantos estão a três passos de distância e assim por diante. Lembrem de anotar no relatório quanto tempo foi necessário para processar todas essas distâncias. Vocês conseguem estimar o tempo de processamento assintótico do programa que vocês escreveram em termos de número de nós  $n$  e de arestas  $e$  da componente gigante?

## Referências

- [SW11] Robert Sedgewick and Kevin Wayne. *Algorithms*. Addison-Wesley Professional, 4th edition, 2011.