

# Exercício de Programação 3:

## Componentes conexas do grafo dos encontros

27 de Abril de 2020

### 1 Introdução Téorica

Vimos que podemos implementar um grafo de três maneiras diferentes e que essa escolha muda a eficiência em termos de tempo de processamento das operações básicas em grafos e espaço na memória. Para testar nossos novos conhecimentos, construímos o grafo dos encontros entre entrevistados na pesquisa Origem/Destino e calculamos o histograma do número de entrevistados que frequentam os mesmos lugares.

Na última aula cobrimos as páginas 529 a 534 e de 543 a 547 da quarta edição do livro *Algorithms* [SW11]. Vimos uma técnica recursiva simples para “passear” por um grafo chamada *busca em profundidade*. Essa técnica é eficaz para extrair informações importantes sobre o grafo que estamos estudando. Em particular, com uma busca em profundidade somos capazes de testar se um grafo é conexo, ou seja, se entre quaisquer dois nós existe um caminho entre eles. Caso o grafo não seja conexo, uma sequência de buscas em profundidade pode nos indicar quantas *componentes conexas* o grafo possui.

Em nosso exemplo, uma pessoa infecta outras apenas quando há um encontro entre elas. Assim, uma pessoa nunca infectará outra que esteja fora de sua componente conexa. Dá a importância de estudar essa propriedade.

## 2 Tarefa

A terceira tarefa consiste em implementar a busca em profundidade para computar o número e o tamanho das componentes conexas no grafo construído na tarefa anterior. O relatório deve conter uma tabela indicando quantos entrevistados estão isolados, quantas componentes existem de tamanho 2, quantas existem de tamanho 3 etc. Façam uma tabela e não um gráfico e omitam os tamanhos em que não há nenhuma componente.

Provavelmente haverá uma componenete muito maior do que as demais. Essa é chamada de *componente gigante*. No nosso caso, quanto maior a componente gigante, mais pessoas estão suscetíveis a serem contaminadas. Dos cenários considerados pelos grupos, qual será que tem maior componente gigante?

## Referências

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