

# TP 01 - Trabalho Prático 01

## Algoritmos I

Samuel Fantini Braga  
2018046637

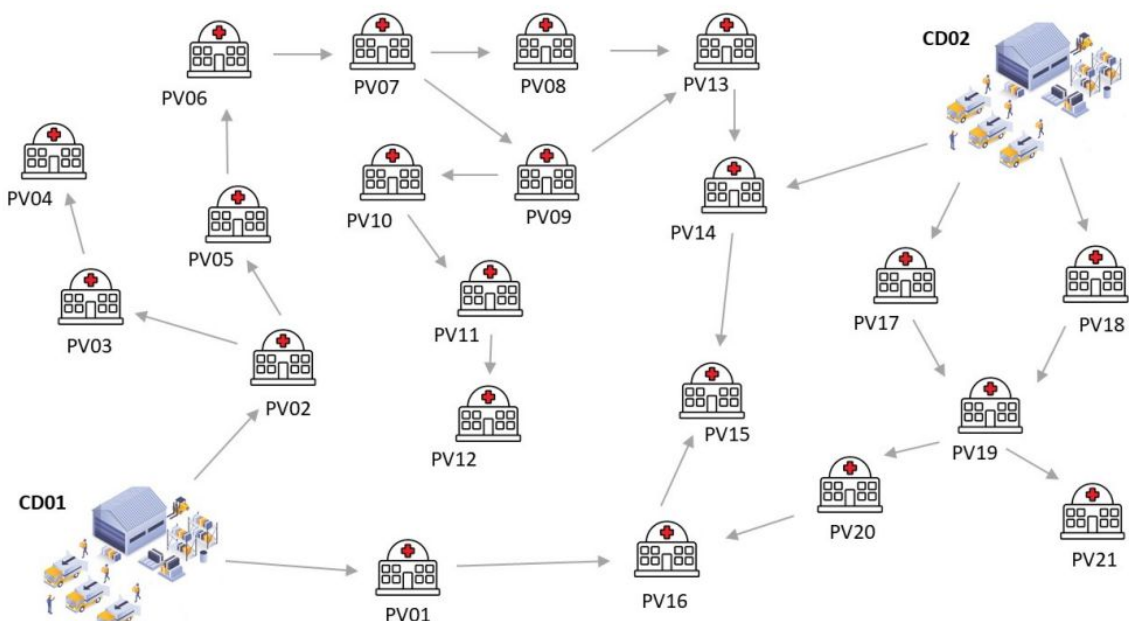
### Modelagem computacional do problema

O problema apresentado consiste na definição de rotas como sendo um caminho, com direção definida iniciando em um centro de distribuição e passando por um ou mais postos de vacinação, uma única vez, até chegar a um posto final do qual não é possível alcançar um outro posto, ou a um posto que foi visitado anteriormente.

Foi especificado uma regra de negócio, onde a mercadoria transportada tem uma temperatura inicial. À medida que a mercadoria transitar entres os postos de vacinação é acrescido uma temperatura a essa mercadoria. Dessa forma, a mercadoria não pode atingir um valor limite. Portanto, existe um limite de transição entre os postos de vacinação.

É necessário também, identificar se no processo de distribuição a rota especificada contém algum ciclo.

### Exemplo da distribuição



## Entrada e saída de dados

### Exemplo das linhas da entrada

```
C P X // Centros de distribuição, postos de vacinação e incremento de temperatura.  
C1 // Linhas que determinam as ligações de cada centro de distribuição  
...  
P1 // Linhas de cada posto de vacinação  
...
```

### Exemplo da saída

```
R // Quantidade de postos de vacinação alcançados.  
PV PV ... // a lista dos R postos de vacinação alcançados, ordenada de forma crescente  
           // pelo código do PV  
C // 1 caso haja alguma rota que percorra um mesmo posto mais de uma vez ou 0, caso contrário
```

## Exemplo prático

### Exemplo prático da entrada

```
2 21 15  
1 2 // c1  
14 17 18 // c2  
16 // p1  
3 5 // p2  
4  
0  
6  
7  
8 9  
13  
10 13  
11  
12  
0  
14  
15  
0  
15  
19  
19  
20 21  
16  
0 // p21
```

### Exemplo prático da saída

```
10 // Qtde de postos alcançados  
1 2 3 5 14 15 16 17 18 19 // Lista dos PVs alcançados  
0 // Não há rotas que passem em um mesmo posto de vacinação mais de uma vez
```

## Estruturas de dados e algoritmos utilizados:

Eu identifiquei que a relação entre os postos de vacinação se mantém independentemente dos postos de vacinação. Dessa forma, chamei essa relação de estrutura interna. Essa estrutura interna representa um grafo de relações.

Portanto eu utilizei uma estrutura de grafo para a resolução dos problemas dessa estrutura interna.

Para os centros de distribuições eu criei um objeto para representá-los. Nesse objeto em específico os postos de vacinação onde eles estão conectados. Dessa forma, eu consigo adicionar N centros de distribuições sem alterar a estrutura interna dos postos de vacinação.

Para a resolução do problema eu implementei o algoritmo de busca em profundidade (depth-first search - DFS). Dessa forma, adicionei apenas uma verificação de temperatura para garantir a continuidade do algoritmo. Onde, para cada nó do grafo é armazenado a sua temperatura e, para cada salto é realizado o acréscimo da temperatura. Como armazenei a temperatura de cada nó, quando o algoritmo retorna para nós anteriores, com caminhos possíveis, o algoritmo continua da mesma forma.

Com isso temos:

1. Criar a quantidade de centros de distribuições necessários;
2. Atribuir os postos de vacinação relacionados a cada centro de distribuição;
3. Criar o grafo da estrutura dos postos de vacinação e suas relações internas;
4. Em seguida, percorrer os caminhos disponíveis para cada centro de distribuição;
5. Coletar os resultados;
6. Apresentar os resultados;

## Lógica para identificar ciclos:

No método *DFSRecursive* adicionei uma verificação para a existência de ciclo. Caso o ciclo exista é alterado o valor de uma propriedade do grafo. Quando buscamos essa propriedade do grafo apenas verificamos se existe o ciclo ou não.

## Problema não solucionados:

Como é feito todos os caminhos possíveis para cada posto de vacinação vinculado com o centro de distribuição temos uma solução muito carregada de valores. Portanto, filtramos e ordenamos nossa solução antes de retornar o resultado para a função main.

## Análise de complexidade de tempo assintótica da solução:

1. Construção dos centros de distribuição  $O(n)$ , onde  $n$  é o número de centros de distribuição -  $O(n)$

2. Adicionar as relações entre centro de distribuição e postos de vacinação  $O(nm)$ , onde  $n$  é o número de centros de distribuição e  $m$  é o número de postos de vacinação relacionados ao centro de distribuição -  $O(nm)$
3. Adicionar as relações internas entre postos de vacinação  $O(nm)$ , onde  $n$  é o número de postos de vacinação e  $m$  o número de postos de vacinação relacionados ao posto de vacinação  $n$ .
4. Realização do algoritmo DFS:
  - a. É realizado a primeira chamada  $O(nm)$ , onde  $n$  é o número de centros de distribuição e  $m$  o número de postos de vacinação vinculados ao centro de distribuição  $n$ .
  - b. No algoritmo DFS temos  $O(v+e)$ , onde  $v$  é o número de nós e  $e$  o número de relações

## Compilação e execução:

Compilador: g++

### Make:

make - build  
make test - execução de testes  
make clean - remoção dos binários gerados

### Execução:

./tp1