Distribuição de vacinas contra o vírus Sars-CoV-2

Marcelo Medeiros de Lima - 2018047030

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)

Abstract. *The article addresses a problem in the distribution of vaccines against the Sars-CoV-2 [2] virus (coronavirus). With the use of Directed graphs [4] and the Breadth-first search [3] (BFS), an approach will be presented to distribute vaccines from each laboratory according to their temperature conservation rules. In accordance with the distribution centers and vaccination posts availability, a list of posts that can be used to apply the vaccine in question is drawn up.*

Resumo. *O artigo aborda um problema de distribuição de vacinas contra o Sars-CoV-2 [2] virus (coronavirus). Com o uso de Grafos dirigidos [4] e do algoritmo de Busca em largura [3] (BFS), será apresentada uma abordagem para se distribuir as vacinas de cada laboratório segundo suas regras de conservação de temperatura. De acordo a disposição dos centros de distribuição e postos de vacinação disponíveis é elaborada uma lista dos postos que poderão ser usados para aplicação da vacina em questão.*

1. Introdução

A Secretaria de Saúde de uma capital brasileira realizou uma grande compra de vacinas de um fabricante. Estas vacinas devem ser mantidas e transportadas a uma temperatura máxima de -60 °C, sob o risco de perder sua eficácia. O fabricante permite a retirada das vacinas em seus centros de distribuição (CD), os quais possuem super refrigeradores e mantêm as vacinas à temperatura de -90 °C . Ao retirar as vacinas de um centro de distribuição, as mesmas devem ser transportadas em embalagens dentro de uma grande caixa térmica com nitrogênio até os postos de vacinação.

Como a Secretaria de Saúde dessa cidade já compra outras vacinas do mesmo fabricante, já existe uma logística de transporte com uma quantidade fixa de caminhões saindo de cada centro de distribuição e rotas já definidas ligando estes centros de distribuição a diversos postos de vacinação espalhados pela cidade.

2. Distribuição da vacina

Durante a entrega das vacinas, no trajeto entre dois postos de vacinação vizinhos, ou entre o centro de distribuição e o primeiro posto da rota, estima-se que haja um incremento da temperatura interna da caixa térmica das vacinas de X°C, devido ao calor trocado entre a caixa térmica e o ambiente do caminhão durante o percurso.

Desta forma, a Secretaria de Saúde deseja calcular, entre todos os PVs pertencentes às possíveis rotas (PVs alcançáveis), a quantidade máxima de postos de vacinação que estarão aptos a realizar a vacinação contra o coronavírus, ou seja,

que receberão as vacinas em temperatura abaixo da temperatura máxima indicada pelo fabricante (PVs alcançados), sem que haja alterações na estrutura logística disponível.

Por fim, a Secretaria de Saúde também deseja identificar se há alguma rota que percorra um mesmo posto de vacinação mais de uma vez, para fins de melhoria de sua estrutura logística.

3. Modelagem computacional

3.1. Centros de distribuição e postos de vacinação

A estrutura de dados escolhida é o grafo dirigido, que representa nos vértices os centros de distribuição e postos de vacinação. Um grafo dirigido é um grafo em que a aresta tem direção. Ou seja, os nós são pares ordenados na definição de cada aresta.

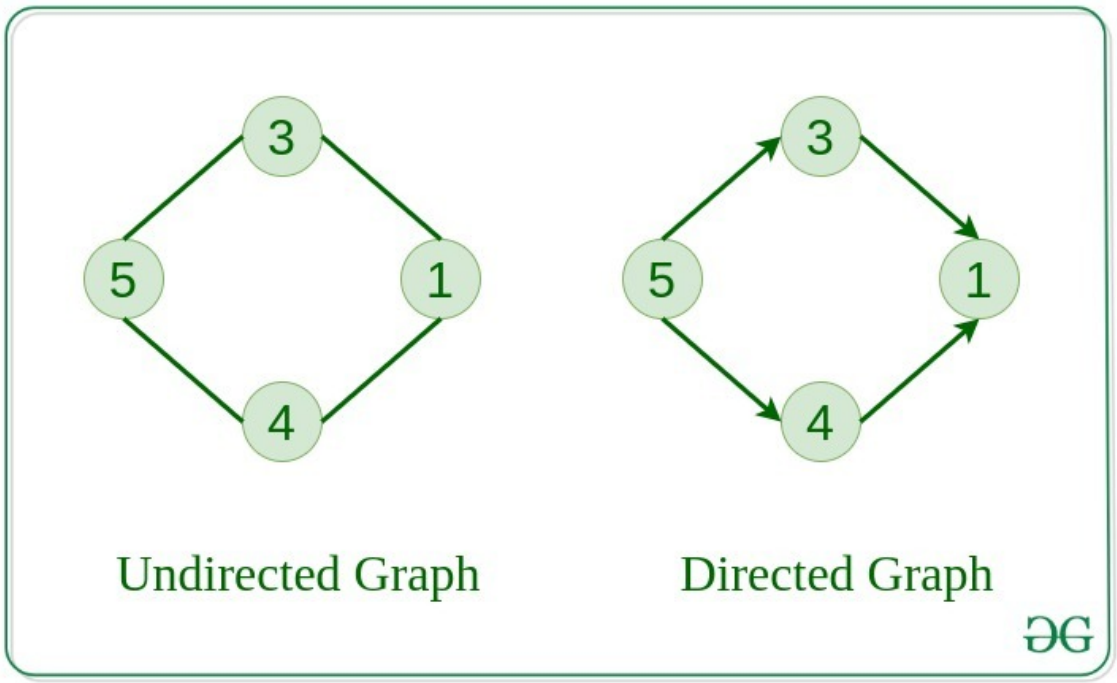

Figura 1. Diferença entre grafos dirigidos e não dirigidos [1]

3.2. Transporte

O transporte das vacinas é feito por uma busca em largura no grafo. A partir de cada centro de distribuição é feito uma busca em largura usando o algoritmo BFS [3]. Com o cálculo de queda de temperatura a cada posto de vacinação que o transporte passa é possível calcular quantas camadas o transporte consegue chegar sem que a vacina perca a eficácia pela queda de temperatura abaixo do exigido.

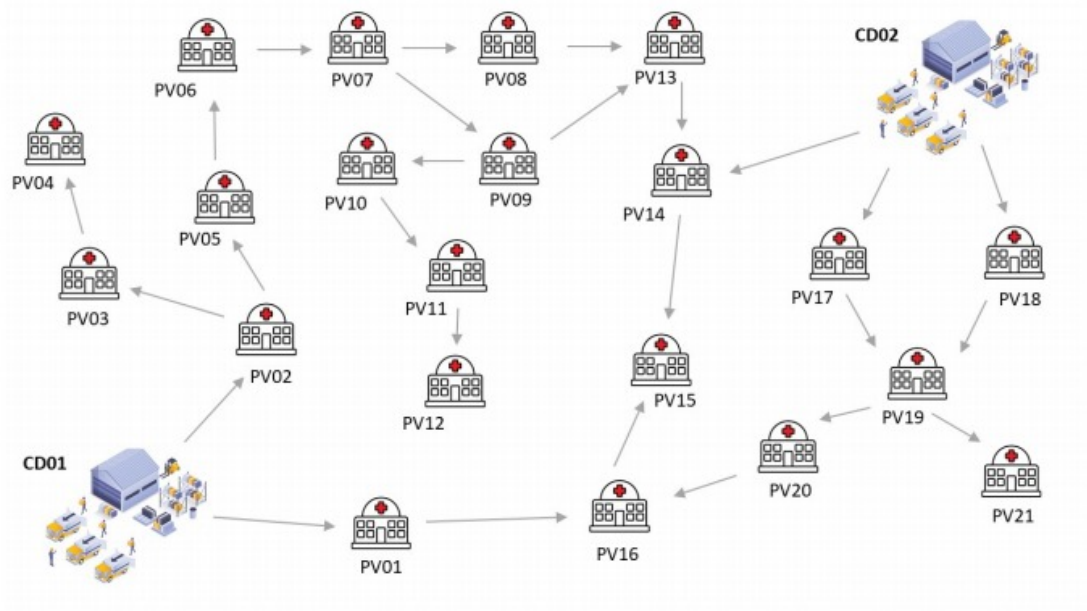

Figura 2. Exemplo de grafo dirigido com os centros e postos

3.3. Complexidade assintótica

A complexidade assintótica do algoritmo é $O(CD^2 + CD * PV)$. Tendo o trecho que procura ciclos no grafo complexidade $O(CD + PV)$ e sabendo que a procura por ciclos ocorre em CD Grafos, ou seja CD vezes, temos $O(CD * (CD + PV)) = O(CD^2 + CD * PV)$

4. Compilação e execução

Para compilar o projeto, existe um makefile na raiz do projeto do CLion (IDE usada no projeto). Para executá-lo, basta digitar make dentro desta pasta. Para executar o projeto basta digitar

```
./tp01 < "nomeDoarquivo".txt (sem as aspas)
```

dentro da mesma pasta para rodar o executável. Foi criada uma pasta de testes para fins de organização do projeto, então caso queira seguir exatamente o exemplo dado, recomenda-se que o arquivo teste de entrada seja movido para essa pasta na raiz do projeto. Caso não consiga executar o arquivo executável gerado, o comando

```
sudo chmod +x tp01
```

deve resolver o problema. Sendo os argumentos:

- **< "nomeDoarquivo".txt >**: Arquivo de entrada com o a quantidade de centros de distribuição, postos de vacinação, queda da temperatura a cada trajeto, além das ligações entre os centros de distribuição e postos de vacinação.

5. Implementação

5.1. Linguagem

Todo o algoritmo e os exemplos de código usados aqui foram desenvolvidos em C++17.

5.2. Grafo

```
class Graph {
    // No. of vertices
    int V;
    // Pointer to an array containing adjacency lists
    std::list<int> *adj;
    // used by isCyclic()
    bool isCyclicUtil(int v, bool visited[], bool *rs);
public:
    // Constructor
    explicit Graph(int V);
    // To add an edge to graph
    void addEdge(int v, int w);
    // Returns true if there is a cycle in this graph
    bool isCyclic();
    void BFS(
            int s,
            std::vector<int> &visitedPosts,
            int layersLimit,
            Graph *CDGraph
            );
};
```

A estrutura mais importante do algoritmo é o grafo em si. A classe do grafo armazena o número de vértices do grafo, além do ponteiro que armazena as listas de adjacências. A partir da dos métodos da classe é possível adicionar arestas ao grafo, verificar se há um ciclo no mesmo e realizar a busca em largura com o BFS [3].

Para encontrar os ciclos no grafo basta que se tenha alcançado mais de uma vez o mesmo posto de vacinação num mesmo caminho possível, seguindo as regras de distribuição da vacina.

> *Um ciclo em um grafo é o grafo ou subgrafo em que o grau de todos os vértices é 2.*

5.3. BFS [3]

Busca em largura para um grafo é semelhante a de uma árvore. O único problema aqui é que, ao contrário das árvores, os gráficos podemos ter ciclos, então podemos voltar ao mesmo nó.

No gráfico a seguir, começamos a travessia do vértice 2. Quando chegamos ao vértice 0, procuramos todos os vértices adjacentes dele. 2 também é um vértice adjacente de 0. Se não marcarmos os vértices visitados, 2 será processado novamente e se tornará um processo não finalizado. A amplitude da primeira travessia do gráfico a seguir seria 2, 0, 3, 1.

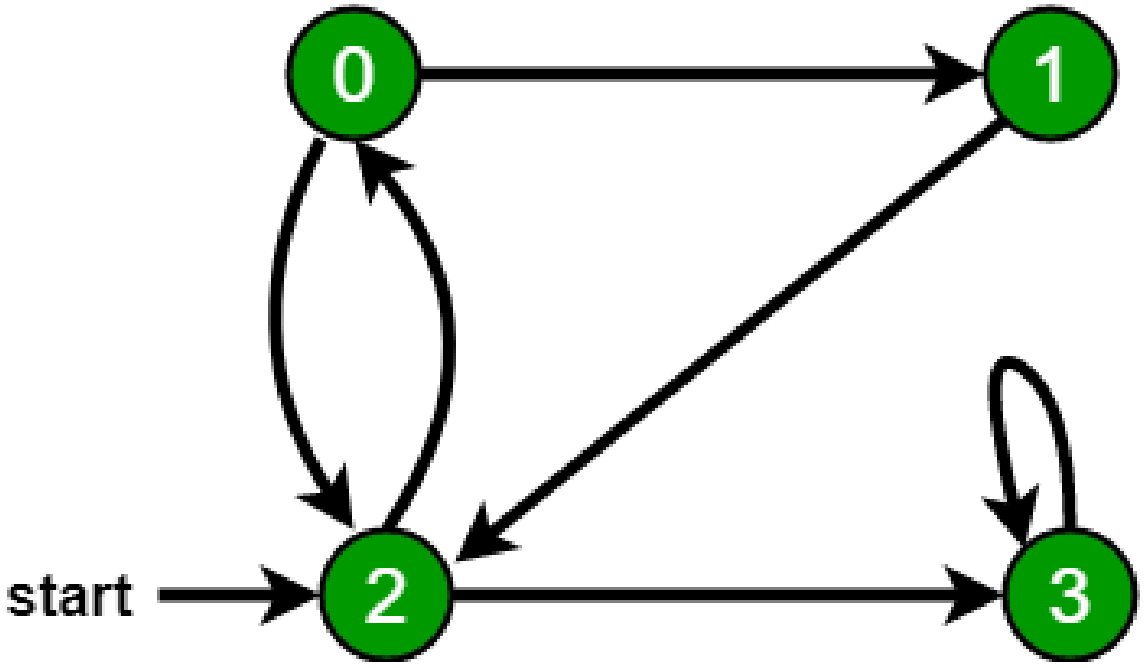

Figura 3. Exemplo de uso do BFS

6. Conclusão

O algoritmo foi implementado com sucesso, sendo utilizado os conhecimentos da disciplina **Algoritmos I** sobre grafos e suas aplicações.

7. Bibliografia

Referências

[1] geeksforgeeks. *Introduction to Graphs*. URL: https://www.geeksforgeeks.org/introduction-to-graphs/. (accessed: 10.01.2021).

[2] UN. *UN Coronavirus Reference*. URL: https://www.un.org/en/coronavirus. (accessed: 22.01.2021).

[3] Wikipedia. *Breadth first search*. URL: https://en.wikipedia.org/wiki/Breadth-first_search. (accessed: 10.01.2021).

[4] Wikipedia. *Directed graph*. URL: https://en.wikipedia.org/wiki/Directed_graph. (accessed: 10.01.2021).