| **Disciplina**  Algoritmos e Estrutura de Dados 2 - [Semestre Letivo: 2022/1] | |
| --- | --- |
| **Nome dos(as) acadêmicos(as)**  1 – Iury Alexandre Alves Bo  2 – Luca Mascarenhas Plaster  3 – Maria Rafaela dos Anjos | **Números de matrícula**  1 – 202103735  2 – 202014610  3 – 202108525 |
| **Turma:** INF0287C | **Professor(a):** Wanderley de Souza Alencar |

**TEMA: ALGORITMO DE BORŮVKA**

I – OBJETIVO

O Algoritmo de Borůvka (ou Algoritmo de Sollin como também é conhecido) é um algoritmo para encontrar uma árvore geradora mínima em um grafo para o qual todos os pesos de arestas sejam distintos. Este algoritmo caracteriza-se pela divisão do grafo original em vários subgrafos para os quais é calculado a *Minimum Spanning Tree* (árvore geradora mínima). Ou seja, no fundo, pode ser considerada uma variação de algoritmos como os de Prim e Kruskal, que são grafos altamente recorrentes para buscar custos mínimos de suas árvores geradoras.

O funcionamento do algoritmo de Borůvka segue os seguintes passos:

1. Para cada vértice, escolher a sua aresta de menor custo. Deste passo poderão resultar um ou mais subgrafos.
2. Caso o passo 1 dê origem a grafos não conectados, considere cada subgrafo gerado no passo anterior como um vértice do grafo final. Então, repita o primeiro passo, encarando cada subgrafo como se fosse um vértice e olhando para as arestas entre esses subgrafos.

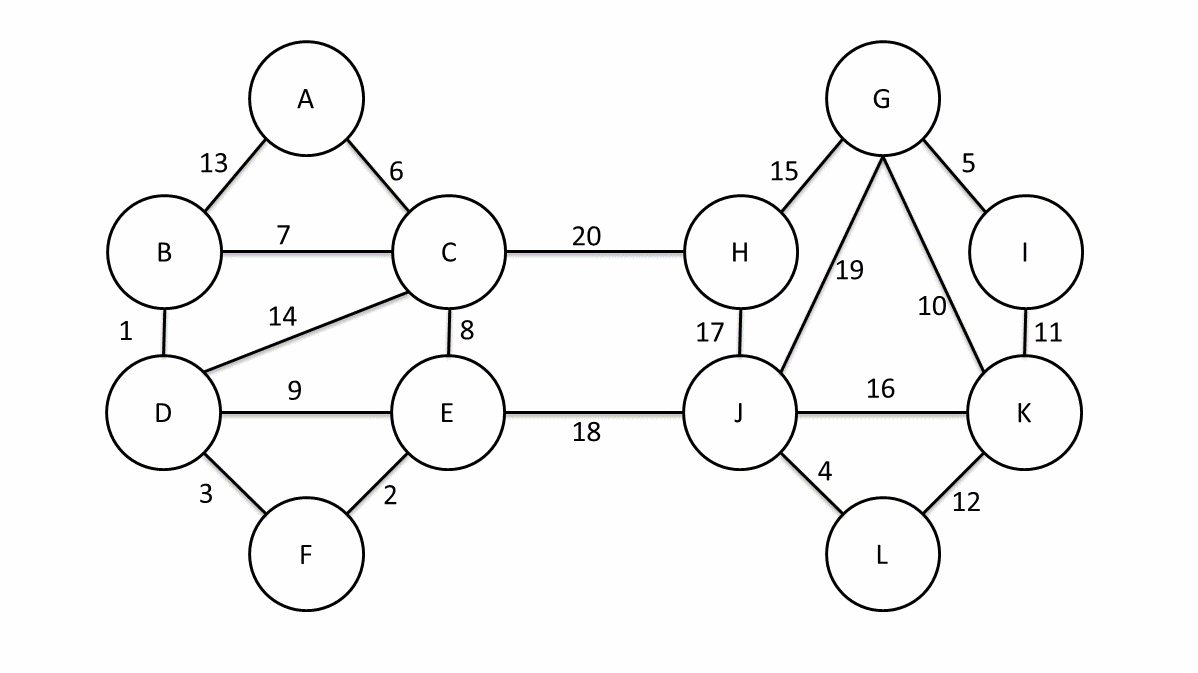


Figura 1: Exemplo prático do algoritmo de Borůvka

Outras curiosidades:

* O algoritmo de Borůvka é o algoritmo de árvore geradora mínima mais antigo descoberto, por Borůvka em 1926, muito antes de os computadores existirem. O algoritmo foi publicado como um método de construção de uma rede elétrica eficiente.
* O problema de mínima arborescência proposta por Borůvka foi registrado por um artigo de uma única página em 1926, tendo como em vista a contribuição para a solução de um problema econômico para a construção de uma rede de energia elétrica por meio de cabos.
* A complexidade de tempo do algoritmo de Borůvka é O(E *log*(V)), que é o mesmo que os algoritmos de Kruskal e Prim.
* O algoritmo de Borůvka é usado como um passo em um algoritmo aleatório mais rápido que funciona em tempo linear O(E). O algoritmo MST de tempo linear esperado é um algoritmo aleatório que calcula a floresta de extensão mínima de um grafo ponderado sem vértices isolados.

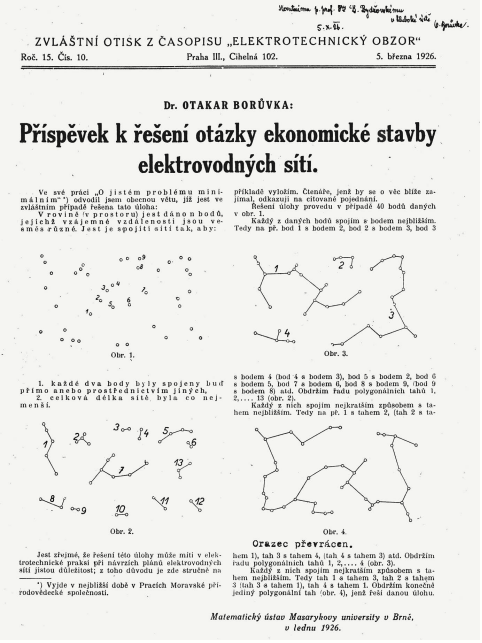
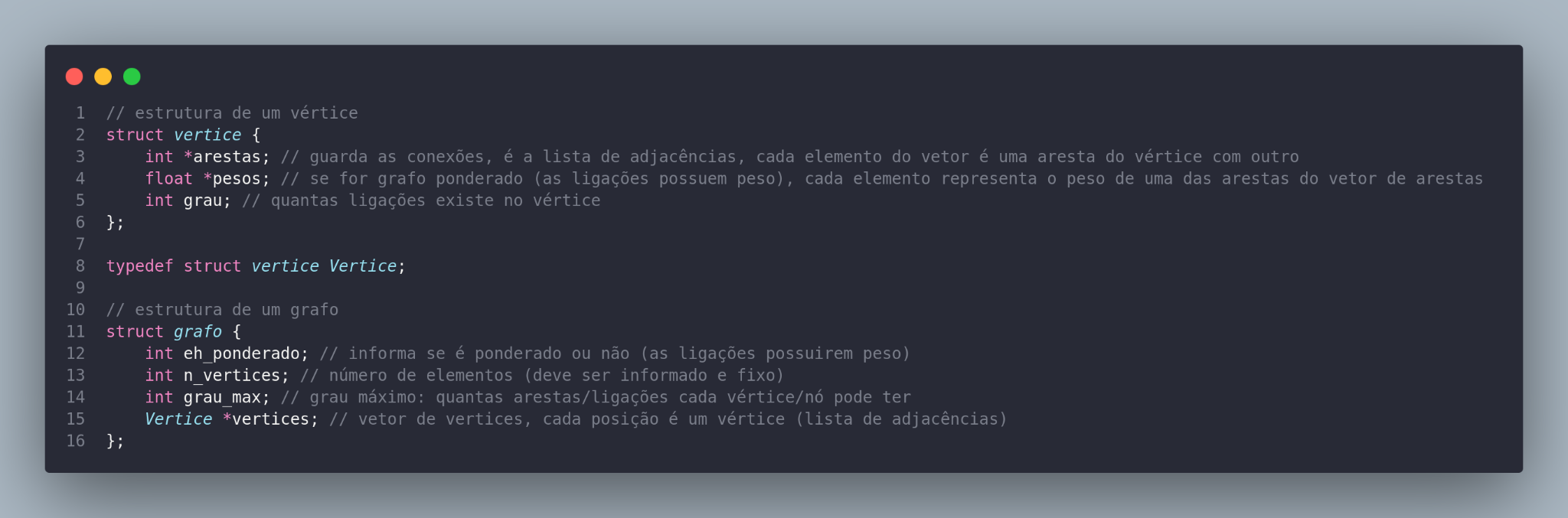


Figura 2: Manchete de pesquisa do Borůvka

II – ALGORITMO

1. Representação computacional do grafo:



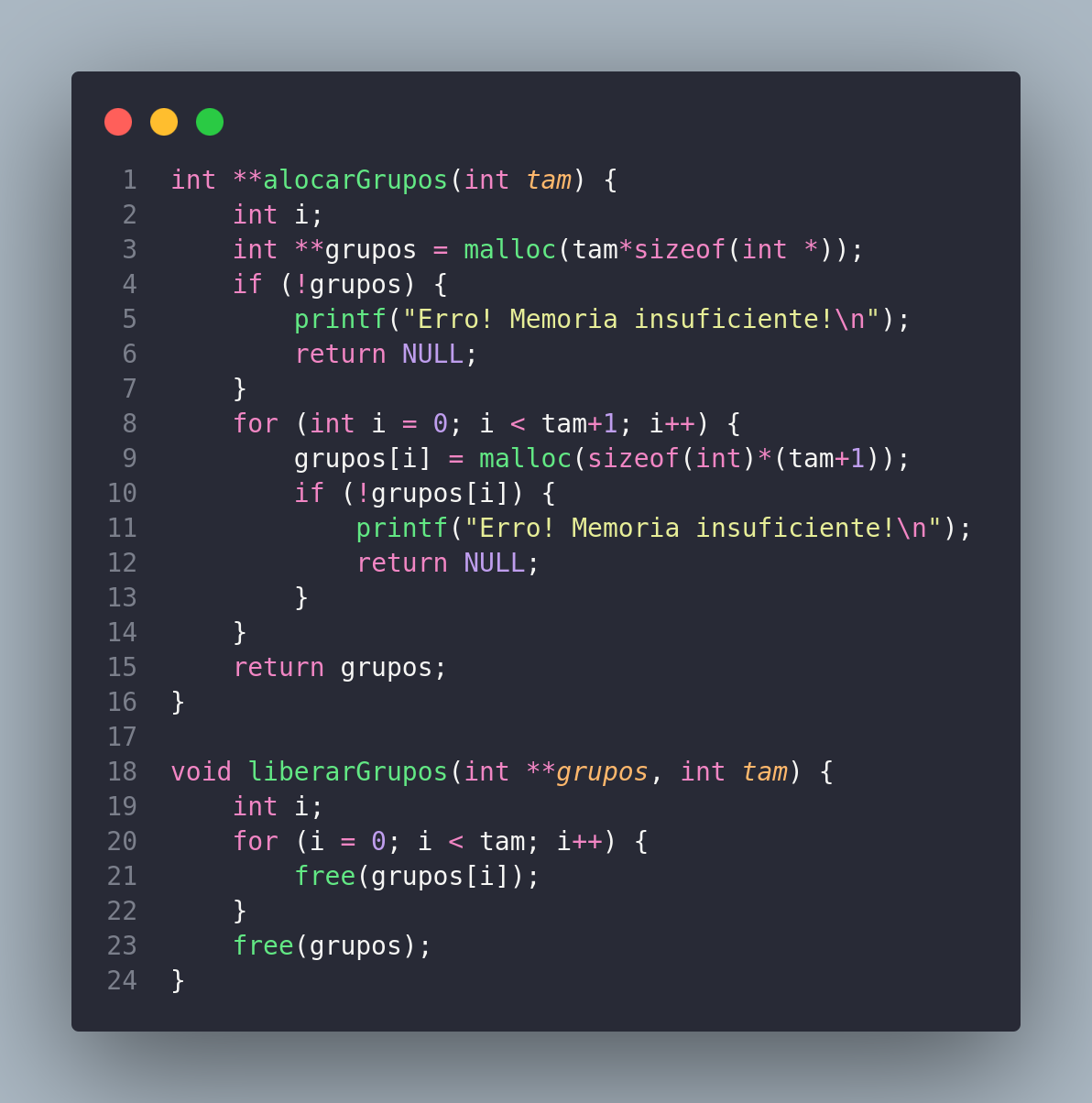
1. Função - Algoritmo de Boruvka - Parte 1:



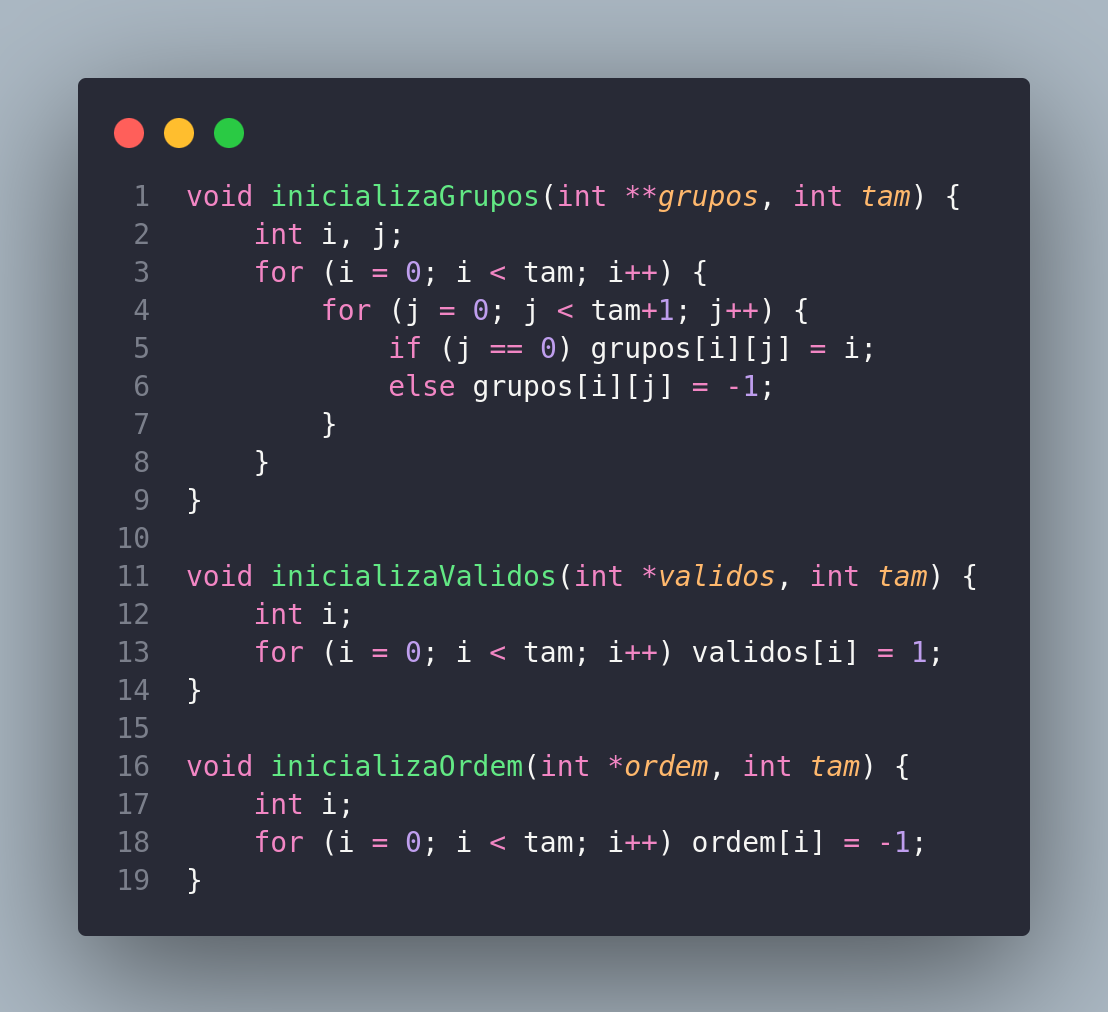
1. Algoritmo de Boruvka - Parte 2:



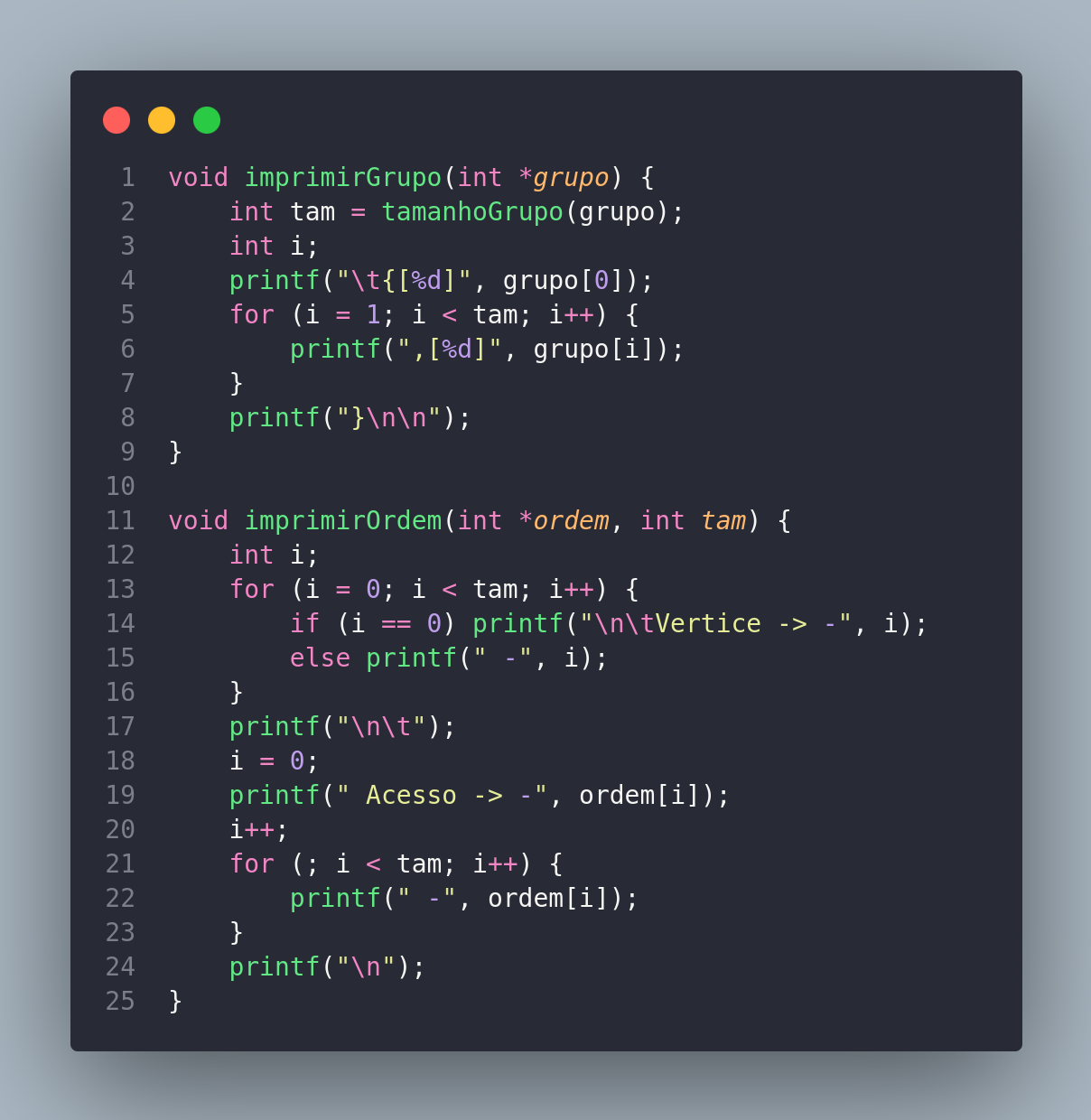
1. Alocar e Liberar grupos:



1. Inicializações:



1. Impressões:



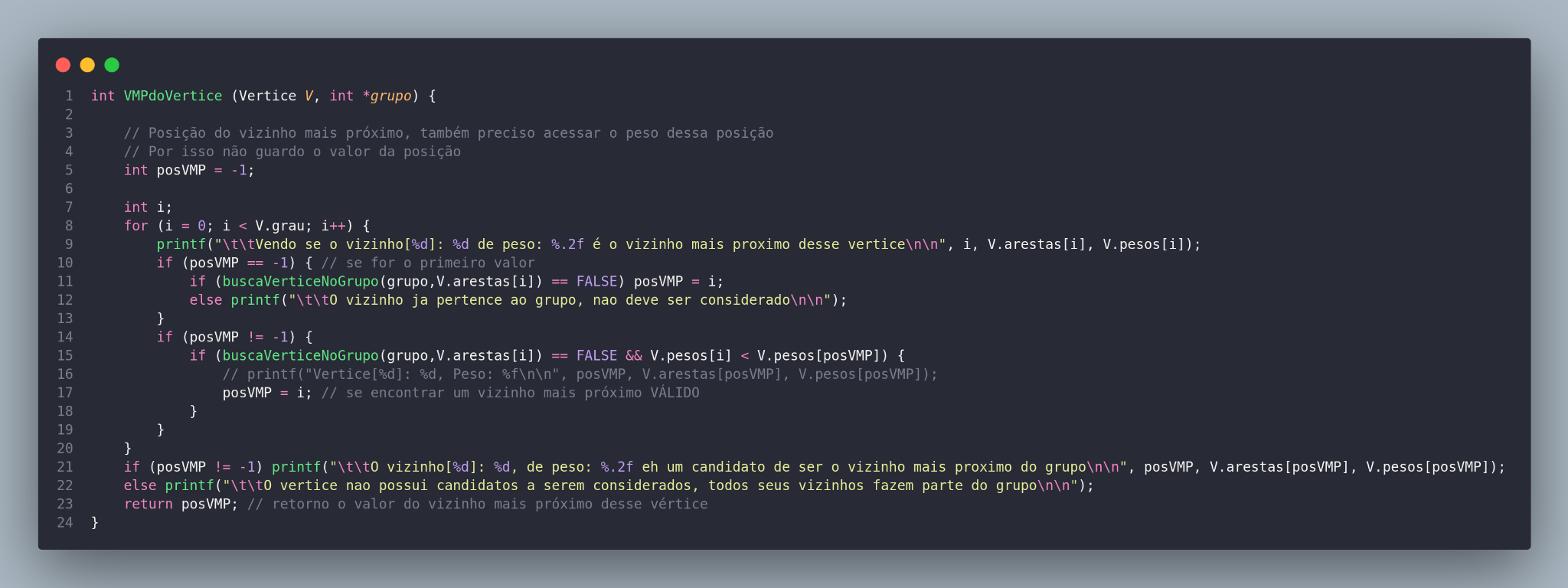
1. Tamanho do grupo:



1. Vizinho mais próximo do grupo:



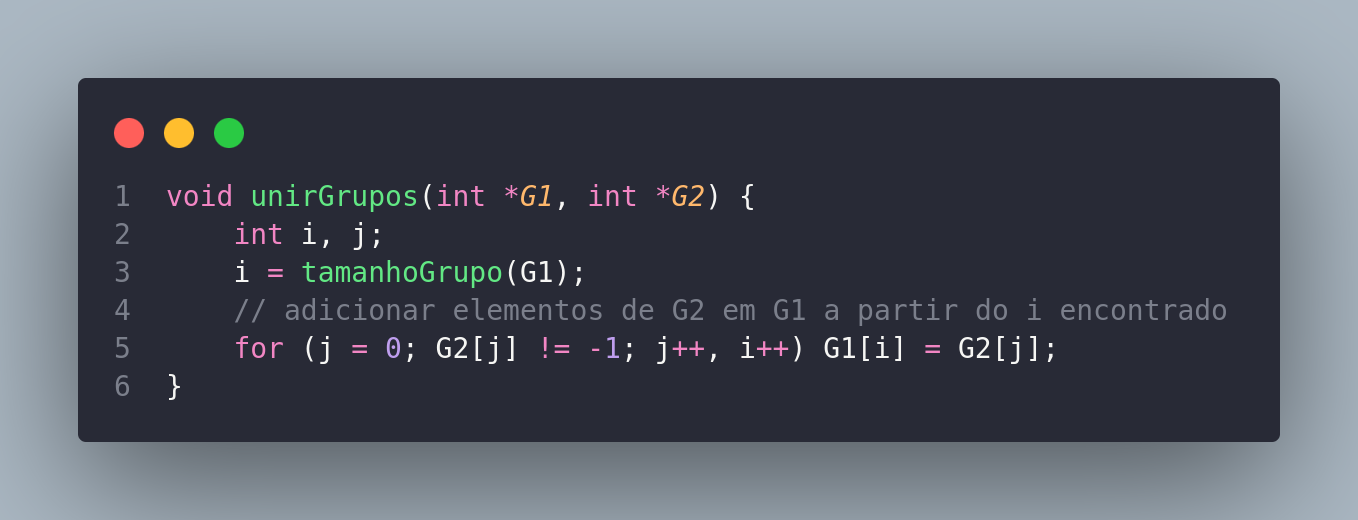
1. Vizinho mais próximo do vértice:



1. Buscar vértice no grupo:



1. Unir grupos:



1. Calcular distância



III – EXEMPLO

Vamos utilizar os passos apresentados no tópico I deste documento para encontrar o menor caminho que percorra todos os vértices do grafo abaixo. Observe que há arestas com o mesmo custo.



Observação: Antes de tudo devemos observar algumas particularidades do grafo, como arestas paralelas e ciclos.



Logo, removeremos o ciclo ligado ao vértice D e a aresta de maior custo (6) que liga C e E, já que são paralelas.



Agora, mantemos a aresta de menor custo referente a cada vértice,mesmo que ela também seja a de menor custo de outro vértice, e desconsideramos as outras.

A aresta de menor custo do vértice A é AB= 7;

A aresta de menor custo do vértice B é BC= 3;

A aresta de menor custo do vértice C também é BC= 3;

A aresta de menor custo do vértice D é DE= 2;

A aresta de menor custo do vértice E também é DE= 2;

A aresta de menor custo do vértice F é EF= 2;



Dessa forma, devemos observar que se o grafo estiver subdividido, deve-se considerar a aresta de menor custo que une os subgrafos. Logo, CE= 3.



Devem existir (n-1) arestas, sendo n o número de vértices. Assim, 6-1=5, que corresponde ao número de arestas do nosso caminho mínimo.



Solução final: 7+3+3+2+2= 17

Supondo que a distância entre um vértice e outro é dada em metros, 17m seria a distância mínima para passar por todos os vértices.

IV – BIBLIOGRAFIA

**ALIESERAJ**, **Animation of Boruvka's algorithm** via Wikipedia, 2012. Disponível em: <[https://en.wikipedia.org/wiki/Borůvka%27s\_algorithm#/media/File:Boruvka's\_algorithm\_(Sollin's\_algorithm)\_Anim.gif](https://en.wikipedia.org/wiki/Bor%C5%AFvka%27s_algorithm#/media/File:Boruvka's_algorithm_(Sollin's_algorithm)_Anim.gif)>;

**INDUSTRIAL21**. **Algoritmo de Boruvka: Teoria dos Grafos (Exemplo Prático)**. Youtube, 2021. Disponível em: <<https://www.youtube.com/watch?v=3cjmO2GCyhU>>;

**O Boruvka**, **Príspevek k reseníotázky ekonomické stavby elektrovodných sítí** (Contribution to the solution of a problem of economical construction of electrical networks), Elektronický obzor 15 (1926), 153–154;

**Pesquisa Operacional**. **Algoritmo de Boruvka: Exemplo Numérico**. Youtube, 2020. Disponível em: <https://www.youtube.com/watch?v=cvxnF1VqVx4>.