

# Atividade Prática 07

## Caminhos Mínimos (Dijkstra)

---

Universidade Tecnológica Federal do Paraná (UTFPR), campus Apucarana  
Curso de Engenharia de Computação  
Disciplina de Estruturas de Dados 1 - EDCO3A  
Prof. Dr. Rafael Gomes Mantovani

---

### Instruções:

- Leia todas as instruções corretamente para poder desenvolver sua atividade/programa;
- Evite plágio (será verificado por meio de ferramentas automatizadas). Faça seu programa com os seus nomes de variáveis e lógica de solução. Plágios identificados anularão as atividades entregues de todos os envolvidos.
- Adicione comentários nos códigos explicando seu raciocínio e sua tomada de decisão. Porém, não exagere nos comentários, pois a própria estrutura do programa deve ser auto-explicativa.
- Salve sua atividade em um arquivo único, com todas as funções e procedimentos desenvolvidos. É esse **arquivo único** que deverá ser enviado ao professor.

## 1 Descrição da atividade

Na teoria dos grafos, o **problema do caminho mínimo** consiste na minimização do *custo* de travessia de um grafo entre dois nós (ou vértices); custo esse dado pela soma dos pesos de cada aresta percorrida. Formalmente, dado um grafo valorado:

$$f : A \longrightarrow \mathbb{R} \quad (1)$$

e dado qualquer elemento  $v$  de  $V$ , o problema dos caminhos mínimos consiste em encontrar um caminho  $P$  de  $v$  para  $v'$ , tal que:

$$\sum_{p \in P} f(p) \quad (2)$$

que é o mínimo entre todos os caminhos conectando os vértices. Existem várias vertentes para problemas de caminho mínimo, cada uma adequada a um conjunto de problemas diferente:

- problema de único destino: consiste em determinar o menor caminho entre cada um dos vértices do grafo e um vértice de destino especificado;
- problema de única origem: consiste em determinar o menor caminho entre um dado vértice e todos os demais no grafo;
- problema de origem-destino: consiste em determinar o menor caminho entre um par de vértices fornecido;
- problemas de todos os pares: consiste em determinar o menor caminho entre cada par de vértices presentes no grafo.

Os algoritmos especializados em solucionar o problema do caminho mínimo são eventualmente chamados de *algoritmos de busca de caminhos*. Entre os algoritmos dessa classe, um dos mais conhecidos é o **algoritmo de Dijkstra**, que resolve um problema com um vértice-fonte em grafos cujas arestas tenham peso maior ou igual a zero. Sem reduzir o desempenho, esse algoritmo é capaz de determinar o caminho mínimo, partindo de um vértice de início  $v$  para todos os outros vértices do grafo.

O problema de caminho mínimo é um dos problemas intensamente estudados e usados em diversas áreas do saber, como Engenharia de Transportes, Pesquisa Operacional, Ciência da Computação e Inteligência Artificial. Isso decorre do seu potencial de aplicação a inúmeros problemas práticos que ocorrem em transportes, logística, redes de computadores e de telecomunicações, entre outros.

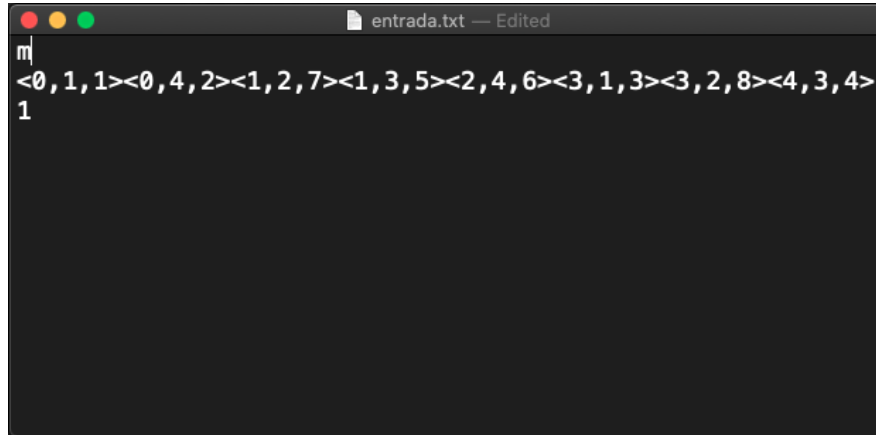
Sendo, assim, sua tarefa é implementar um programa que apresente o resultado do algoritmo de Dijkstra, encontrando caminhos mínimos em grafos ponderados com pesos não negativos a partir de uma origem.

## 2 Entradas e saídas do programa

A Figura 1a mostra um exemplo de arquivo de entrada. O programa recebe alguns parâmetros, nesta sequência:

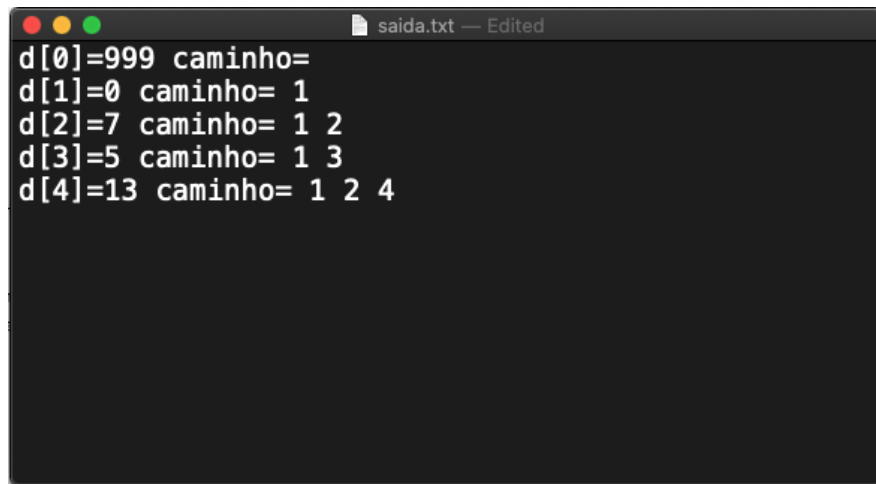
- a representação a ser usada - 'm' para matriz de adjacência, e 'l' para lista de adjacência;
- as arestas de um grafo dirigido com pesos não negativos nas arestas. O formato é: <origem,destino,peso>; e
- o vértice de origem dos caminhos.

Com base nessa entrada, o programa deve gerar um arquivo de saída (Fig 1b) com os 'custos' dos caminhos mínimos do vértice de origem até os vértices alcançáveis a partir dele, bem como os vértices que formam tais caminhos.



```
m
<0,1,1><0,4,2><1,2,7><1,3,5><2,4,6><3,1,3><3,2,8><4,3,4>
1
```

(a) Exemplo de arquivo de entrada para Dijkstra.



```
d[0]=999 caminho=
d[1]=0 caminho= 1
d[2]=7 caminho= 1 2
d[3]=5 caminho= 1 3
d[4]=13 caminho= 1 2 4
```

(b) Exemplo de arquivo de saída para Dijkstra.

Figura 1: Valores de entrada e correspondentes arquivos de saída gerado pelo programa.

### 3 Algumas observações importantes

- a quantidade de vértices não é diretamente informada. Portanto é sua tarefa descobrir com base nos valores do arquivo de entrada;
- os  $n$  vértices são rotulados com os valores de  $(0, n-1)$ ;
- como o grafo é dirigido, a indicação no arquivo de entrada de uma aresta  $\langle 0,1,3 \rangle$  deve ser interpretada como  $0 \rightarrow 1$  com peso 3;
- O valor 'infinito' será definido como 9999;
- Os vértices que não são alcançáveis a partir do vértice de origem têm o curso 'infinito' e nenhum vértice formando o caminho.

## 4 Orientações gerais

Além da funcionalidade desejada, implementar também o controle de erros, para lidar com exceções que possam ocorrer, como por exemplo:

- quantidade de parâmetros na linha de comando é diferente do ideal;
- problemas nas aberturas dos arquivos de entrada e saída;
- arquivo de entrada vazio (sem informação);
- arquivo de entrada fora do padrão esperado;
- grafo com arestas de peso negativo;
- etc.

### 4.1 Dados para envio da atividade

Ao elaborar seu programa, crie um único arquivo fonte (.c) seguindo o padrão de nome especificado:

ED1-<ANO>-<SEMESTRE>-AT07-Dijkstra-<NOME>.c

Exemplo:

ED1-2021-2-AT07-Dijkstra-RafaelMantovani.c

A entrega da atividade será via Moodle: o link será disponibilizado na página da disciplina.

## 5 Links úteis

Arquivos em C:

- <https://www.inf.pucrs.br/~pinho/LaproI/Arquivos/Arquivos.htm>
- <https://www.geeksforgeeks.org/basics-file-handling-c/>
- <https://www.programiz.com/c-programming/c-file-input-output>

Argumentos de Linha de comando (argc e argv):

- [https://www.tutorialspoint.com/cprogramming/c\\_command\\_line\\_arguments.htm](https://www.tutorialspoint.com/cprogramming/c_command_line_arguments.htm)
- <http://linguagemc.com.br/argumentos-em-linha-de-comando/>
- [http://www.univasf.edu.br/~marcelo.linder/arquivos\\_pc/aulas/aula19.pdf](http://www.univasf.edu.br/~marcelo.linder/arquivos_pc/aulas/aula19.pdf)
- [http://www.inf.ufpr.br/cursos/ci067/Docs/NotasAula/notas-31\\_Argumentos\\_linha\\_comando.html](http://www.inf.ufpr.br/cursos/ci067/Docs/NotasAula/notas-31_Argumentos_linha_comando.html)
- <http://www.dca.fee.unicamp.br/cursos/EA876/apostila/HTML/node145.html>

## Referências

- [1] Thomas H. Cormen,; Ronald Rivest; Charles E. Leiserson; Clifford Stein. Algoritmos - Teoria e Prática - 3<sup>a</sup> Ed. Elsevier - Campus, 2012.
- [2] Nivio Ziviani. Projeto de algoritmos com implementações: em Pascal e C. Pioneira, 1999.
- [3] Adam Drozdek. Estrutura De Dados E Algoritmos Em C++. Cengage, 2010.