

# Algoritmos e Estruturas de Dados 2

## Terceiro Trabalho Prático (Primeira Versão)

**Entrega Sugerida: 17/01/2020**

1º Semestre 2020 - DC-UFSCar

## 1 Introdução

No quarto Trabalho Prático (TP03) será solicitada a entrega de um programa que solucione o problema apresentado na próxima seção. O arquivo entregue deve seguir os itens abaixo:

- O TP03 deverá ser feito individualmente e plágio não será tolerado;
- O TP03 deve ser entregue no run codes (<https://run.codes>) em um arquivo contendo código em uma das seguintes linguagens (C, C++, Java) e com um cabeçalho com as informações do estudante (nome, curso, RA);
- Cada estudante deve se cadastrar no run codes (<https://run.codes>) informando Nome Completo, escolhendo “UFSCar - Universidade Federal de São Carlos” no campo Universidade e colocando seu RA no campo Núm. Matrícula. Depois de cadastrado, basta logar no run codes e se matricular na disciplina “1001490 - Algoritmos e Estruturas de Dados 2” usando o Código de Matrícula JVBG.
- Não é permitida a utilização de bibliotecas ou funções built-in que implementam soluções para os problemas estudados na disciplina. Ex: `list.sort()`, `string.find()`, etc;
- Se escolher linguagem C, compile o seu TP usando o compilador GCC com flags `-Wall -pedantic -O2 -Wno-unused-result`, pois warnings podem impedir o código de funcionar no run codes ainda que funcionem no seu computador.

## 2 Crush

Poucas pessoas sabem qual foi a motivação de Edsger Dijkstra para inventar seu famoso algoritmo de caminhos mais curtos. Certo dia, ele estava estudando matemática quando sua crush mandou uma mensagem no Telegram

dizendo “Oi, Ed! Estou em casa sem fazer nada e meus pais saíram. Mas eles voltam logo, vem rápido!”. Para ter certeza que chegaria logo na casa da crush, ele precisou achar um caminho que fosse rápido até lá. Daí surgiu seu algoritmo.

É verdade. Eu estava lá. Eu era o celular do Dijkstra.

Como você já deve saber, Dijkstra, apesar de ser um dos melhores computadores de todos os tempos, não é o melhor dos programadores. Ajude ele a fazer um programa que receba um grafo ponderado e direcionado como entrada e devolva a menor distância entre sua casa e a casa de sua crush. Assim, ele não perderá oportunidades únicas!

### **Entrada:**

Cada entrada contém um único caso de teste. A primeira linha contém dois inteiros, o número de vértices  $V$  ( $5 \leq V \leq 1000$ ) e o número de arestas  $E$  ( $5 \leq E < 10^6$ ) do grafo. As próximas  $E$  linhas contêm as informações das arestas. Especificamente, cada linha tem três números inteiros,  $A$  ( $0 \leq A < V$ ),  $B$  ( $0 \leq B < V$ ) e  $W$  ( $2 \leq W \leq 100$ ), indicando que há uma aresta (direcionada) de  $A$  para  $B$  com peso  $W$ . A casa de Dijkstra se encontra no vértice 0 e a casa da crush se encontra no vértice  $V-1$ . É garantido que há um caminho entre esses dois vértices.

### **Saída:**

Seu programa deve imprimir apenas uma linha contendo um inteiro, indicando qual é a distância entre a casa do nosso amigo Edsger e seu (futuro) moção.

### **Exemplo de entrada**

```
8 13
0 3 7
1 0 3
2 1 3
2 4 6
2 5 6
3 5 2
4 7 5
5 1 6
```

5 6 3  
5 7 6  
6 2 9  
6 7 1  
7 6 3

**Saída esperada para esse exemplo**

13