

Instruções sobre o Exercício 4

1001323 — Algoritmos em Grafos

Cândida Nunes da Silva

1^o Semestre de 2022

1 Problema – Orquídeas

A empresa DComp-So (Distribuição e Comércio de Orquídeas Multicoloridas Phalaenopsis de Sorocaba) possui diversas estufas para o cultivo de orquídeas espalhadas pela região de Sorocaba. Para cada cor de orquídea Phalaenopsis certas condições ambientais são mais favoráveis para o cultivo de belas pencas de orquídeas. Existem ligações dos locais das estufas para uma rede de pontos de distribuição das orquídeas da empresa para todos os estabelecimentos quem vendem suas orquídeas para o consumidor final. De toda estufa é possível chegar em qualquer centro de distribuição, embora possa ser necessário passar por outros centros de distribuição no caminho. Cada uma das ligações possui um custo de transporte que é proporcional à distância e outros fatores logísticos. Os negócios da empresa estão indo de vento em popa ultimamente, e o volume das vendas aumentou muito em pouco tempo. Surgiu então a necessidade de otimizar o custo das entregas das orquídeas aos centros de distribuição. Para isso, o gerente de vendas busca saber para cada centro de distribuição qual a estufa ideal para fazer o abastecimento, levando em consideração o custo de transporte para se utilizar cada uma das ligações disponíveis.

Sua implementação da solução deve ter complexidade limitada a $O(n^2 \log n)$.

2 Entrada

A entrada deve ser lida da entrada padrão (teclado). A primeira linha da entrada contém três inteiros N , M e K . O valor N ($1 \leq N \leq 10^4$) representa quantos são no total as estufas e os pontos de distribuição. Cada um desses pontos é representado por um inteiro entre 0 e $N - 1$. As estufas são representadas pelos K menores números e os centros de distribuição pelos demais, isto é, os números $K, \dots, N - 1$. O valor M ($1 \leq M \leq 10^5$) representa o número total de ligações entre estufas e centros de distribuição e entre os centros de distribuição. As M linhas subsequentes contêm pares de inteiros A e B ($0 \leq A, B \leq N - 1$) e um número C ($1 \leq C \leq 10^4$) que representa o custo de transporte por essa ligação.

3 Saída

A saída deve ser escrita na saída padrão (terminal). A saída consiste em uma única linha, contendo $N - K$ inteiros, representando os números das estufas que atendem com menor custo cada um dos centros de distribuição. Os $N - K$ valores devem ser apresentados em ordem crescente do número do centro de distribuição, isto é, na ordem $K, \dots, N - 1$. Caso mais de uma estufa atenda um centro de distribuição com o mesmo custo, deve ser escolhida a estufa com menor número.

4 Exemplo

Entrada	Saída
8 10 3 0 6 35 1 3 15 2 4 12 3 4 23 3 5 25 3 6 34 3 7 10 4 5 10 4 7 5 5 6 15	1 2 2 0 2

Entrada	Saída
6 8 2 0 2 20 0 3 15 1 5 20 2 1 6 2 4 17 3 4 10 4 1 22 4 5 14	1 0 1 1

5 Desenvolvimento e Apresentação

Cada aluno deve implementar a sua solução individual. A implementação da solução do problema deve ser em C em arquivo único. O nome do arquivo deve estar na forma “ex04-nomesn.c”, onde “nomesn” representa o primeiro nome do aluno seguido das iniciais de seu sobrenome. Note que todas as letras são minúsculas e o separador é “-” (hífen) e não “_” (underscore).

O juiz online verificará seu programa comparando para cada um dos casos de teste se a saída gerada pelo seu programa é igual à saída esperada. É **imprescindível** que o **algoritmo** implementado esteja correto, isto é, retorne a solução esperada para **qualquer** entrada. É **desejável** que a implementação seja eficiente.

6 Ambiente de Execução e Testes

O programa deve ser compilável em ambiente Unix com gcc. Sugere-se que os testes também sejam feitos em ambiente Unix. Deve-se esperar que a entrada seja dada na entrada padrão (teclado) e

não por leitura do arquivo de testes. Da mesma forma, a saída deve ser impressa na saída padrão (terminal), e não em arquivo.

A motivação dessa exigência é apenas simplificar a implementação de entrada e saída, permitindo o uso das funções `scanf` e `printf` da biblioteca padrão para leitura e escrita dos dados, sem precisar manipular arquivos.

Por outro lado, é evidente que efetivamente entrar dados no teclado é muito trabalhoso. Em ambiente Unix, é possível usar redirecionamento de entrada na linha de comando de execução para contornar esse problema. Supondo que o nome do arquivo executável seja análogo ao arquivo fonte, e “ex04.in” seja o arquivo com os casos de teste, a linha de comando:

```
shell$ ./ex04-nomesn < ex04.in
```

executa o programa para todos os casos de teste de uma só vez, retornando todas as saídas em sequência para o terminal. Novamente, pode-se usar o redirecionamento de saída na linha de comando para escrever a saída em um arquivo de saída de nome, por exemplo, “ex04.my.out”. A respectiva linha de comando seria:

```
shell$ ./ex04-nomesn < ex04.in > ex04.my.out
```

Após a execução, a comparação pode ser feita usando o comando `diff` do Unix. Por exemplo, se o arquivo “ex04.out” contém as saídas esperadas, a linha de comando:

```
shell$ diff ex04.out ex04.my.out
```

serve para comparar automaticamente os dois arquivos, retornando nada caso sejam idênticos e as linhas onde há discrepâncias caso contrário.

7 Notas

As notas serão baseadas na correção da solução implementada, clareza do código fonte e eficiência da solução.

Trabalhos que não atendam aos requisitos mínimos expressos neste documento de forma a inviabilizar o teste do programa receberão nota ZERO. Em particular, receberá nota ZERO todo programa que:

- não compila em ambiente Unix;
- dá erro de execução;
- não usa entrada e saída padrão para leitura e escrita.
- apresentar fortes indícios de cola.