

PUC-Rio  
Departamento de Informática  
Prof. Marcus Vinicius S. Poggi de Aragão  
Período: 2016.1  
16 de maio de 2016  
Horário: 2as-feiras e 4as-feiras de 9 às 11 horas

Data da Entrega: 20 de junho de 2016

## Estruturas Discretas (INF 1631)

### 2º Trabalho de Implementação

#### Descrição

Este trabalho prático consiste em desenvolver códigos para diferentes algoritmos e estruturas de dados para resolver os problemas descritos abaixo.

Adicionalmente, a análise do desempenho das implementações destes algoritmos com respeito ao tempo de CPU deve ser feita para permitir um entendimento do uso de recursos computacionais.

O desenvolvimento destes códigos e a análise devem seguir os seguintes roteiros:

- Descrever os algoritmos informalmente.
- Demonstrar o entendimento do algoritmo explicando, em detalhe, o resultado que o algoritmo deve obter e justificá-lo.
- Explicar a fundamentação do algoritmo e justificar a sua corretude apresentando a prova por indução matemática que leva ao algoritmo.
- Apresentar as tabelas dos tempos de execução obtidos pelos algoritmos sobre as instâncias testadas.
- Documente o arquivo contendo o código fonte de modo que cada passo do algoritmo esteja devidamente identificado e deixe claro como este passo é executado.
- Para a medida de tempo de CPU das execuções utilize as funções disponíveis no link correspondente na página do curso, um exemplo de utilização é apresentado. Quando o tempo de CPU for inferior à 5 segundos, faça uma repetição da execução tantas vezes quantas forem necessárias para que o tempo ultrapasse 5 s (faça um while), conte quantas foram as execuções e reporte a média.

A corretude código deverá ser testada sobre um conjunto de instâncias. O trabalho entregue deve conter:

- Um documento contendo o roteiro de desenvolvimento dos algoritmos (e dos códigos), os itens pedidos acima, respostas a perguntas do trabalho, comentários e análises sobre a implementação e os testes realizados (**entrega digital e impressa**).

- **O envio de um e-mail para [poggi@inf.puc-rio.br](mailto:poggi@inf.puc-rio.br) com o ASSUNTO (ou SUBJECT) ED161T2 é obrigatório.** Este email deve conter todos arquivos correspondentes ao trabalho. O NÃO ENVIO DESTE E-MAIL IMPLICA QUE O TRABALHO NÃO SERÁ CONSIDERADO. Caso seja envie mais de um email com este assunto, identifique com a versão. Exemplo: ED161T2 V2.
- O trabalho pode ser feito em grupo de até 3 alunos.

### 1. *Walk of a King*

Considere um tabuleiro de xadrez e um rei que está inicialmente na posição  $(1, 1)$  (as posições do tabuleiro são representadas por  $(i, j)$  onde  $1 \leq i \leq 8$  e  $1 \leq j \leq 8$ ). Para cada posição do tabuleiro estão associados um prêmio  $p_{ij}$  e um consumo  $q_{ij}$  (o prêmio pode ser em USD(!!) e o consumo em litros de gasolina, por exemplo). Os prêmios e os consumos assumem somente valores positivos. O rei tem inicialmente  $Q$  unidades para consumir e pode passar quantas vezes quiser em cada posição do tabuleiro e a cada vez receber o prêmio e, naturalmente, consumir os seus recursos. Ao final (do passeio) **o rei tem que estar de volta na posição  $(1, 1)$ .**

- (a) Apresente um algoritmo para determinar o caminho que o rei deve fazer para obter o maior total possível em prêmios.
- (b) Desenvolva um código para o seu algoritmo e teste para um arquivo de dados com a estrutura apresentada abaixo.

#### **Descrição do Arquivo de Entrada**

A primeira linha contém um inteiro  $Q$  que indica o número de unidades que o rei tem para consumir. Seguem dois blocos de 8 linhas cada uma com 8 inteiros positivos. O primeiro bloco corresponde ao consumo de cada posição do tabuleiro, enquanto que o segundo bloco corresponde ao prêmios atribuídos às posições.

A posição  $(1,1)$  é a primeira das oito na primeira linha de cada bloco. Na mesma linha se encontram as posições  $(1,2)$ ,  $(1,3)$ ,  $\dots$   $(1,8)$ . A linha seguinte se inicia com o valor da posição  $(2,1)$ , a linha seguinte com o valor da posição  $(3,1)$  e assim por diante. A última linha de cada bloco contém valores referentes às posições  $(8,1)$ ,  $(8,2)$ ,  $\dots$ ,  $(8,8)$ . Naturalmente esta linha é precedida pela linha  $(7,1)$ ,  $(7,2)$ ,  $\dots$ ,  $(7,8)$ .

O arquivo de dados possui múltiplas instâncias e termina quando o valor de  $Q$  é zero.

#### **Descrição do Arquivo de Saída**

Para cada entrada do arquivo acima você deve imprimir o maior prêmio possível, o total consumido, o valor de  $Q$  e a sequência de posições correspondente à caminhada do rei.

## 2. Go Easy

Uma cidade acaba de implantar um novo sistema de cobrança dos transportes públicos urbanos. O sistema de transportes opera com trens e ônibus. A tarifação é feita em unidades de transporte UTs (uma espécie de vale transporte com valor fixo). Para o ônibus cada viagem é paga. A tarifa para os trens é baseada nas zonas das estações de entrada e de saída do trem. O valor é determinado seguindo as regras:

- (a) Paga-se 2 UTs toda vez que se entra no sistema de transportes, seja em um trem ou em um ônibus.
- (b) Se passageiro pegou um trem, ele paga 4 UTs se as zonas da estações de origem e destino forem diferentes, caso contrário nenhuma UT adicional é paga.
- (c) Para cada ônibus que o passageiro pega ele paga 1 UT.

Um mapa do sistema de transportes é fornecido como entrada. Nele encontra-se uma lista das estações com suas respectivas zonas (que são usadas pelos trens e pelos ônibus, por exemplo, um ponto de ônibus é uma estação sem trem). Encontra-se também os itinerários dos trens e dos ônibus dados por sequências de estações.

Os trens e os ônibus movem-se nos dois sentidos em cada itinerário e nenhum trem ou ônibus passa duas vezes em uma mesma estação durante uma ÚNICA viagem (ao final de cada viagem os passageiros devem descer do trem ou ônibus).

**O objetivo do trabalho é implementar um algoritmo que, dados o mapa do sistema de transportes, os itinerários dos trens e dos ônibus, a origem e o destino do passageiro, encontre os ônibus e trens que o passageiro deve pegar, explicitando onde este deve trocar de ônibus ou de trem, de modo que o custo em UTs seja mínimo.**

Um arquivo contendo várias entradas de dados (os descritos acima) será disponibilizado. Em todas as entradas (mapas) existem meios para se transportar entre qualquer par de estações. Outra observação importante é de que as regras são estritas, se um passageiro pega duas vezes o mesmo ônibus ele paga as duas vezes, se ele entra duas vezes na mesma zona (em viagens diferentes) ele paga as duas vezes.

### Descrição do Arquivo de Entrada

A primeira linha contém dois inteiros  $Z$  e  $S$ , que indicam respectivamente o número de zonas ( $1 \leq Z \leq 30$ ) e o número de estações de trem/ônibus na cidade ( $1 \leq S \leq 100$ ). Cada estação tem um identificador único (de 1 a  $S$ ) pertence a exatamente uma zona. As próximas  $Z$  linhas descrevem as estações que pertencem a cada zona. As linhas começam com um inteiro  $K$  indicando o número de estações na zona, seguido por  $K$  inteiros representando as estações nesta zona. Em seguida, o arquivo contém um linha com dois inteiros  $T$  e  $B$ , representando respectivamente o número de itinerários de trem ( $1 \leq T \leq 50$ ) e ônibus ( $1 \leq B \leq 50$ ). Nas  $T + B$  linhas seguintes estão os  $T$  itinerários de trem seguidos dos  $B$  itinerários de ônibus. A descrição de cada itinerário começa com um inteiro  $L$  ( $2 \leq L \leq S$ ) indicando o número de estações no itinerário, seguido por  $L$  inteiros representando a sequência das estações que definem o itinerário. Finalmente, vem um linha com dois inteiros  $X$  e  $Y$  ( $1 \leq X \leq S$ ,  $1 \leq Y \leq S$  e  $X \neq Y$ ) indicando as estações de origem e de destino do passageiro respectivamente. O final o arquivo é indicado pelo inteiros  $Z$  e  $S$  com valor zero.

### Descrição do Arquivo de Saída

Para cada entrada do arquivo acima você deve imprimir o custo mínimo do transporte entre a origem e o destino dados assim como as ações do passageiro para obter esse custo.