TP 02 - Trabalho Prático 02

Algoritmos I

Entrega: 17/12/2021

1 Objetivos do trabalho

O objetivo deste trabalho é modelar o problema computacional descrito a seguir utilizando uma estrutura de dados que permita resolvê-lo de forma eficiente com os algoritmos estudados nesta disciplina.

Serão fornecidos alguns casos de teste bem como a resposta esperada para que o aluno possa verificar a corretude de seu algoritmo. Não obstante, recomenda-se que o aluno crie casos de teste adicionais a fim de validar sua própria implementação.

O código-fonte da solução e uma documentação sucinta (relatório contendo não mais do que 5 páginas) deverão ser submetidos via *moodle* até a data limite de 17/12/2021. A especificação do conteúdo do relatório e linguagens de programação aceitas serão detalhadas nas seções subsequentes.

2 Definição do problema

A solução proposta no trabalho prático 1 para a empresa varejista foi um sucesso. O algoritmo desenvolvido por você para agendamento de clientes durante o evento da Black Friday auxiliou na tomada de decisão e no atendimento aos principais clientes, aumentando o volume de vendas da empresa como um todo. A solução de encontrar listas de prioridades com os principais clientes de cada loja foi muito elogiada e o seu trabalho foi reconhecido. Por isso, o presidente da varejista resolveu contratar novamente o seu trabalho para solucionar um problema no processo de delivery de vendas online, trazendo melhorias nos processos e redução de custos.

Uma das estratégias para crescimento da rede de lojas e maior fidelização dos clientes é melhorar o processo de entrega das compras realizadas online, modalidade de venda ainda pouco explorada pelo varejista. Para isso, um dos grandes desafios é otimizar o deslocamento de produtos entre lojas da empresa. Esta otimização poderá trazer redução no custo de transporte, redução de estoque disponível em cada loja e ganho de escala. Visando isto, a equipe de negócios da empresa varejista decidiu que deveria ser implementado um algoritmo para selecionar os principais trajetos entre lojas da empresa de forma que o custo total do somatório de todos eles seja o menor possível. Por trajeto entende-se como a rota direta em linha reta entre duas lojas A e B, sem haver uma loja intermediária.

A empresa possui N lojas espalhadas pela cidade e, por pertencer a uma grande capital do país, é possível estabelecer um trajeto entre qualquer par de lojas. Existem também três modos de transporte que podem ser utilizados no processo de entrega: por meio de motocicletas, com a utilização de caminhões ou por meio de drones. O transporte por drone é o mais barato e, portanto, podemos considerá-lo como de custo 0 (zero), mas devido ao alto custo de aquisição existem somente D ($1 \leq D < N$) drones disponíveis e, por restrição da tecnologia, cada drone só pode transportar mercadoria para outra loja que também utilize o meio de transporte por drones. Cada drone tem a capacidade de atender por dia somente a um trajeto

entre duas lojas e todas as lojas podem operar com este tipo de transporte. O transporte por motocicleta tem uma limitação de autonomia diária que faz com que este meio só possa ser utilizado em trajetos de até K km ($1 \leq K \leq 10000$), trajetos mais longos devem ser realizados por meio de caminhões. O custo do km rodado das motocicletas tem um valor igual a M e o custo do km rodado de caminhões tem o valor C ($1 \leq M \leq C, C \leq 10$). Não há limite no número de motocicletas e caminhões disponíveis para uso.

O sistema de cadastro da empresa possui as informações de cada loja l, sendo o número de identificação id_l e a localização geográfica (x_l, y_l) ($-10000 <= x_l, y_l <= 10000$) as informações disponíveis e necessárias neste momento.

O objetivo é encontrar uma solução que forneça uma combinação de trajetos de forma que, ao percorrer cada trajeto uma única vez ao dia e em uma única direção, o custo geral de deslocamento seja minimizado. Deve existir pelo menos uma forma de comunicação entre qualquer par de lojas da empresa, mas esta comunicação não deve ser obrigatoriamente direta, podendo ser necessário percorrer vários trajetos para alcançar o destino, mesmo que em dias diferentes. Ou seja, você não precisa se preocupar em garantir que todo par de lojas estejam conectadas no mesmo dia, este planejamento será parte de uma segunda etapa do projeto.

3 O que fazer?

Você deve projetar e codificar um algoritmo que seja capaz de encontrar os melhores trajetos entre as lojas da empresa e qual o melhor meio de transporte a ser utilizado de forma que o custo geral de deslocamento seja minimizado e que exista pelo menos uma forma de comunicação entre quaisquer lojas da empresa.

Este algoritmo deverá:

- Ler o número N de lojas da empresa, o limite máximo K de km utilizado por uma motocicleta em um trajeto entre duas lojas, o número D de drones disponíveis, o custo M do km rodado por motocicletas e o custo C do km rodado por caminhões;
- Ler o id e a localidade geográfica de cada uma das lojas da empresa;
- Encontrar a melhor combinação de trajetos que minimize o custo diário de se percorrer todos os eles uma vez ao dia em direção única mantendo todas as lojas conectadas.
- Definir qual meio de transporte será utilizado em cada trajeto.
- Calcular o orçamento necessário para cada meio de transporte na situação onde cada trajeto selecionado seja percorrida uma vez ao dia em uma única direção.

Considere uma situação onde a empresa possua N lojas, seu programa deverá selecionar uma combinação de trajetos de forma que todas as lojas se mantenham conectadas e, ao percorrer todos os trajetos uma única vez ao dia e em uma única direção, não exista outra combinação com custo total menor que o custo do conjunto de trajetos selecionados. O custo total de um conjunto de trajetos é definido como o somatório do custo de se percorrer cada um deles uma única vez.

A distância entre cada loja deve ser considerada como a distância euclidiana entre elas, que pode ser calculada pela seguinte fórmula, considerando duas lojas l e i: $d(l, i) = \sqrt{(x_i - x_l)^2 + (y_i - y_l)^2}$.

4 Arquivos de entrada

O problema terá como entrada um arquivo contendo as informações como o número N de lojas, o limite K de km máxima para uso de motocicletas, a quantidade D de drones disponíveis, o custo M do km rodado

por motocicleta, o custo C do km rodado por caminhão e as informações das lojas da empresa. Os alunos terão que ler o arquivo e armazená-lo em uma estrutura de dados para utilizar na resolução do problema. O arquivo está organizado da seguinte forma:

A primeira linha irá conter cinco números N, K, D, M, C, sendo os valores correspondentes para o número de lojas, o limite máximo de km para uso de motocicletas, a quantidade de drones disponíveis, o custo do km rodado por motocicleta e o custo do km rodado por caminhão. As próximas N linhas irão conter um par (X_l, Y_l), separados por espaço, correspondente a coordenada de cada loja da empresa. O id da loja deverá ser definido como um número sequencial a partir de 0 definido pela ordem de leitura de cada registro.

5 Saída

A saída deve ser impressa na tela (*stdout*) e seguir o seguinte padrão: Deverá ser impresso um par de números c_m, c_c indicando o custo total da utilização de motos e caminhões, respectivamente. Este custo deverá considerar que cada trajeto encontrado foi percorrido somente uma vez. Sem salto de linha.

Fique atento para imprimir exatamente como foi descrito pois a correção do algoritmo será feita de forma automática.

6 Exemplo do problema

A seguir é apresentado um exemplo:

Dados de Entrada

```
6 1 3 2 3
0 0
0 2
2 0
3 2
2 3
3 3
```

A imagem abaixo ilustra a leitura deste exemplo após realizada a leitura das coordenadas de cada loja e calculada a distância entre elas.

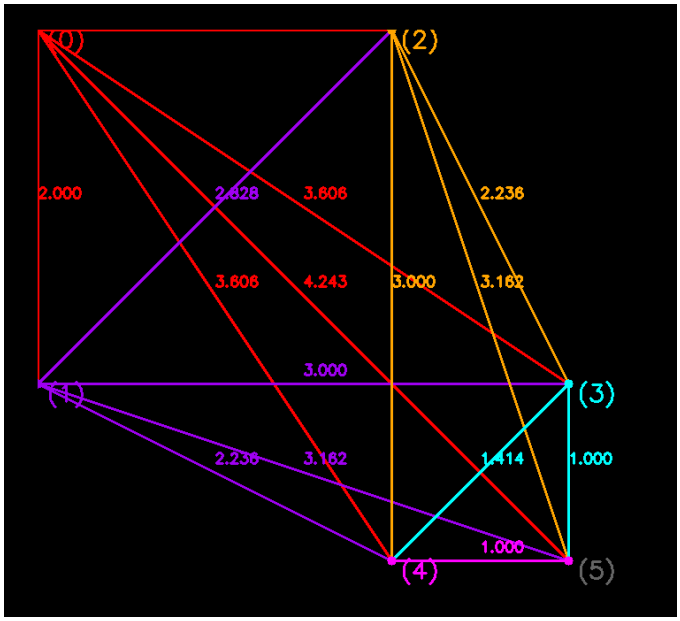

Figura 1: Exemplo de localização das lojas e distância entre cada uma delas

Como existem 3 drones disponíveis, três subconjuntos de lojas se formarão com um drone disponível em uma loja de cada um deles, os mantendo conectados. De acordo com a imagem, os conjuntos são escolhidos considerando a distância mínima que conecta cada um deles, resultando em:

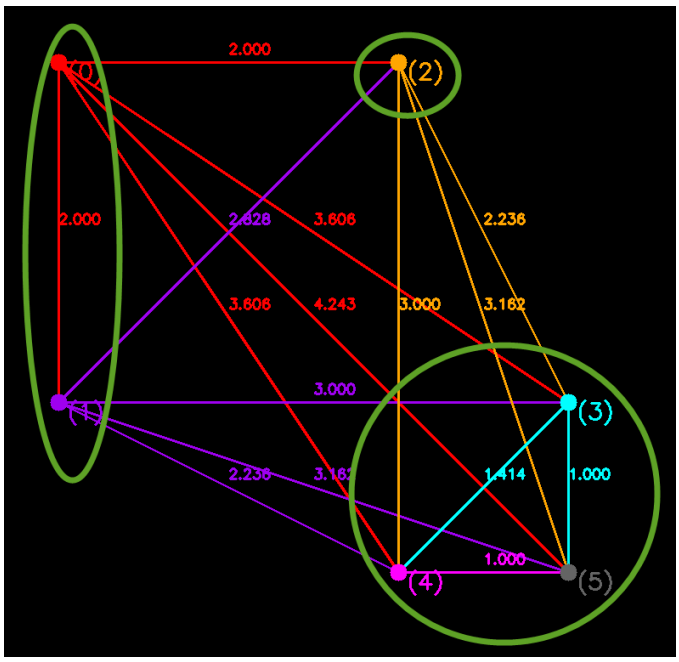

Figura 2: Exemplo de formação de 3 conjuntos de lojas comunicadas

A partir dos dados de entrada, temos $K = 1$, $M = 2$ $C = 3$. Portanto, os deslocamentos que têm uma distância menor que $1km$ entre 2 lojas serão percorridas por motociclista, enquanto os que forem maiores que $1km$ serão percorridas por caminhão.

A distância da loja 3 à 5 é de 1km, por isso se considera o uso de motocicletas. Da loja 4 à 5 é de $1km$ também, onde também será utilizado um motociclista. Da loja 0 a 1 são $2km$, então um caminhão será usado.

O número total de km a serem percorridos por motociclistas é de $2km$ e multiplicando este número por $M = 2$ (custo do motociclista por km) temos como resultado 4. O número total de km a serem percorridos por caminhões é de $2km$ e multiplicando este número por $C = 3$ (custo do caminhão por km) resulta em 6.

Portanto, o resultado final é:

Dados de Saída

```
4.000 6.000
```

7 Exemplo Prático de Entrada e Saída

Arquivo de entrada

```
3 1 1 1 1
0 0
0 1
1 0
```

Exemplo prático da saída

```
2.000 0.000
```

Arquivo de entrada

```
6 1 3 2 3
0 0
0 2
2 0
3 2
2 3
3 3
```

Exemplo prático da saída

```
4.000 6.000
```

Arquivo de entrada

```
19 500 4 7 10
100 256
152 -502
-9999 9999
0 -50
500 458
12 654
-855 -20
-1 -1
0 0
100 -100
-100 100
45 987
5000 -4999
4999 8000
75 89
1000 500
1000 100
25 958
25 9255
```

Exemplo prático da saída

```
18126.335 63961.210
```

8 Especificação das entregas

Você deve submeter um arquivo compacto (zip ou tar.gz) no formato **MATRICULA_NOME** via Moodle contendo:

- todos os arquivos de código-fonte implementados;
- um arquivo *makefile*[1] **que crie um executável com nome tp01**;
  - **ATENÇÃO:** O makefile é para garantir que o código será compilado da forma como vocês implementaram, evitando erros na compilação. É **essencial** que ao digitar "make" na linha de comando dentro da pasta onde reside o arquivo makefile, o mesmo compile o programa e gere um executável chamado **tp01**.
- sua documentação (arquivo pdf).

[1] https://pt.wikibooks.org/wiki/Programar_em_C/Makefiles

Sua documentação deverá ser sucinta e conter não mais do que 5 páginas com o seguinte conteúdo obrigatório:

- Modelagem computacional do problema;
- estruturas de dados e algoritmos utilizados para resolver o problema (pseudo-código da solução impelementada), bem como justificativa para tal escolha. Não transcreva trechos da código-fonte;
- análise de complexidade de tempo assintótica da solução proposta, devidamente justificada.

9 Implementação

9.1 Linguagem, Ambiente e Parâmetros

O seu programa deverá ser implementado na linguagem **C** ou **C++** e deverá fazer uso apenas de funções da biblioteca padrão da linguagem. Trabalhos que utilizem qualquer outra linguagem de programação e/ou que façam uso de bibliotecas que não a padrão não serão aceitos.

O aluno pode implementar seu programa em qualquer ambiente (Windows, Linux, MacOS, etc...), no entanto, deve garantir que seu código compile e rode nas máquinas do DCC (tigre.dcc.ufmg.br ou jaguar.dcc.ufmg.br), pois será neste ambiente que o TP será corrigido. Note que essas máquinas são acessíveis a todos os alunos do DCC com seu login e senha, podendo inclusive ser realizado acesso remoto via ssh. O aluno pode buscar informações no site do CRC (Centro de Recursos Computacionais) do DCC (https://www.crc.dcc.ufmg.br/).

O arquivo da entrada deve ser passado ao seu programa como entrada padrão, através da linha de comando (e.g., *$ ./tp01 < casoTeste01.txt*) e gerar o resultado também na saída padrão (não gerar saída em arquivo).

ATENÇÃO: Não é necessário que o aluno implemente em ambiente Linux. Recomenda-se que o aluno teste seu código nas máquinas previamente especificadas, as quais serão utilizadas para correção do TP, a fim de conferir a funcionalidade, makefile e demais características do código.

9.2 Testes automatizados

A sua implementação passará por um processo de correção automatizado, utilizando além dos casos de testes já disponibilizados, outros exclusivos criados para o processo de correção. O formato da saída de seu programa deve seguir a especificação apresentada nas seções anteriores. Saídas diferentes serão consideradas erro para o programa. O aluno deve certificar-se que seu programa execute corretamente para qualquer entrada válida do problema.

ATENÇÃO: O tempo máximo esperado para execução do programa, dado o tamanho máximo do problema definido em seções anteriores, é de 5 segundos.

9.3 Qualidade do código

Preze pela qualidade do código-fonte, mantendo-o organizado e comentado de modo a facilitar seu entendimento para correção. Caso alguma questão não esteja clara na documentação e no código fonte, a nota do trabalho pode ser penalizada.

10 Critérios para pontuação

A nota final do TP (NF) será composta por dois fatores: fator parcial de implementação (fpi) e fator parcial da documentação (npd). Os critérios adotados para pontuação dos fatores é explicado a seguir.

10.1 Fator parcial de implementação

Serão avaliados quatro aspectos da implementação da solução do problema, conforme a Tabela 1.

Aspecto	**Sigla**	**Valores possíveis**
Compilação no ambiente de correção	co	0 ou 1
Respostas corretas nos casos de teste de correção	ec	0 a 100%
Tempo de execução abaixo do limite	te	0 ou 1
Qualidade do código	qc	0 a 100 %

Tabela 1: Aspectos de avaliação da implementação da solução do problema

O fator parcial de implementação será calculado pela seguinte fórmula:

$$fpi = co \times (ec - 0,15 \times (1 - qc) - 0,15 \times (1 - te))$$

Caso o valor calculado do fator seja menor que zero, ele será considerado igual a zero.

10.2 Fator parcial da documentação

Serão avaliados quatro aspectos da documentação entregue pelo aluno, conforme a Tabela 2.

Aspecto	**Sigla**	**Valores possíveis**
Apresentação (formato, clareza, objetividade)	ap	0 a 100%
Modelagem computacional	mc	0 a 100%
Descrição da solução	ds	0 a 100%
Análise de complexidade de tempo assintótica	at	0 a 100 %

Tabela 2: Aspectos de avaliação da documentação

O fator parcial de documentação será calculado pela seguinte fórmula:

$$fpd = 0,4 \times mc + 0,4 \times ds + 0,2 \times at - 0,25 \times (1 - ap)$$

Caso o valor calculado do fator seja menor que zero, ele será considerado igual a zero.

10.3 Nota final do TP

A nota final do trabalho prático será obtida pela equação a seguir:

$$NF = 10 \times (0,6 \times fpi + 0,4 \times fpd)$$

É importante ressaltar que é obrigatória a entrega do código fonte da solução e documentação. Na ausência de um desses elementos, a nota do trabalho prático será considerada igual a zero, pois não haverá possibilidade de avaliar adequadamente o trabalho realizado.

Assim como em todos os trabalhos dessa disciplina é estritamente proibida a cópia parcial ou integral de código-fontes, seja da Internet ou de colegas. Se for identifcado o plágio, o aluno terá a nota zerada e o professor será informado para que as medidas cabíveis sejam tomadas.

ATENÇÃO: Os alunos que submeterem os TPs com atraso, terão a nota final penalizada em termos percentuais de acordo com a seguinte regra: $2^{d-1}/0,16$ (onde d é a quantidade de dias úteis de atraso na entrega do TP)