TP 03 - Trabalho Prático 03

Algoritmos I

Entrega: 23/03/2021

1 Objetivos do trabalho

O objetivo deste trabalho é modelar o problema computacional descrito a seguir visando resolvê-lo de forma eficiente.

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 23/03/2021. A especificação do conteúdo do relatório e linguagens de programação aceitas serão detalhadas nas seções subsequentes.

2 Definição do problema

João da Silva é um jovem universitário que gosta de conhecer outras cidades ao redor do mundo. Antes de agendar suas viagens ele planeja todos os custos de passagens, hospedagem, alimentação e transporte, pois geralmente seu orçamento é bem apertado, e também planeja os pontos turísticos que deseja conhecer.

O próximo destino de João é Nova York, mas infelizmente devido à recente valorização do dólar, ele terá de economizar no valor que havia planejado para o transporte. Para isso, ele resolveu utilizar o metrô da *Big Apple*, que recentemente passou por uma revisão no formato de cobrança dos bilhetes. Como o ponto de destino pode estar a várias estações de distância, João poderá precisar pegar vários metrôs subsequentemente.

O novo sistema de cobrança dos bilhetes, que está em fase de teste de aceitação, funciona da seguinte forma: o passageiro paga por cada transição de linha de metrô realizada (definida como escala) e o bilhete para cada escala pode ter preço diferente. Foi implantada uma politica de descontos cumulativos (D_i, desconto cumulativo para i-ésima escala em diante) nas escalas realizadas dentro de um intervalo de **T** minutos, até um limite máximo de **D** escalas nesse intervalo. Transcorridos os **T** minutos desde o início da primeira escala (tempo >= T), o passageiro perde o desconto acumulado e inicia novamente a progressão de descontos na próxima escala que fizer. Caso dentro do intervalo T, o passageiro realize mais do que D escalas, a partir da escala D+1, o passageiro não faz mais jus ao desconto cumulativo, pagando então o preço cheio do bilhete até o término do tempo T.

Tendo estudado a fundo o novo sistema de cobrança do metrô, para chegar a cada atração turística que deseja visitar, João fez uma lista das escalas que devem ser realizadas sequencialmente no metrô. Ele incluiu nessa lista, o tempo de translado de cada escala e o preço cheio de cada bilhete. Agora, ele precisa da sua ajuda para calcular o custo mínimo para realizar o transporte para cada atração turística. Note que João deve realizar as escalas exatamente na ordem listada. No entanto, ele aceita esperar o tempo que

for necessário nas transições de linhas de metrô (apesar de passar metrô a cada minuto) de forma a obter o menor custo de transporte possível em sua estadia em Nova York.

3 Exemplo do problema

Cada caso de teste conterá na primeira linha três inteiros **N** ($1 \leq N \leq 10^4$) que representa a quantidade de escalas a serem percorridas até chegar no destino final, **D** ($2 \leq D \leq 10^2$) (quantidade máxima de escalas com descontos cumulativos no intervalo T) e **T** ($1 \leq T \leq 10^3$), representando o intervalo máximo para aplicação dos descontos acumulados (transcorrido esse prazo desde a primeira escala, o passageiro perde o desconto acumulado e inicia novamente a progressão de descontos na próxima escala que fizer).

Na segunda linha, aparecem D inteiros ($5 \leq D_i \leq 10^3$), representando o desconto percentual cumulativo para cada escala em diante no valor do bilhete, até a D-ésima escala no intervalo T.

Nas próximas N linhas, existirão dois inteiros, E e C, representando o tempo da viagem e o custo do bilhete dessa escala, em que ($1 \leq E, C \leq 10^3$).

O arquivo de saída deverá conter uma única linha representando o menor custo possível para que João consiga chegar ao seu destino final (**truncado em duas casas decimais**).

Exemplo das linhas da entrada

```
N D T // Qtde de escalas, quantidade máxima de escalas com descontos cumulativos
         no intervalo T e tempo máximo para aplicação de descontos.
d_1 d_2 d_i // Quantidade de desconto fornecido na escala D_i
t_1 c_1 // Tempo gasto no translado da escala 0 e o preço do bilhete desta escala
t_i c_i
...
```

Exemplo da saída

```
C // Custo mínimo para alcançar o destino final
```

4 Arquivos de entrada e saída

4.1 Primeiro exemplo

Exemplo de entrada

```
3 3 5
0 50 0
4 5
3 10
5 9
```

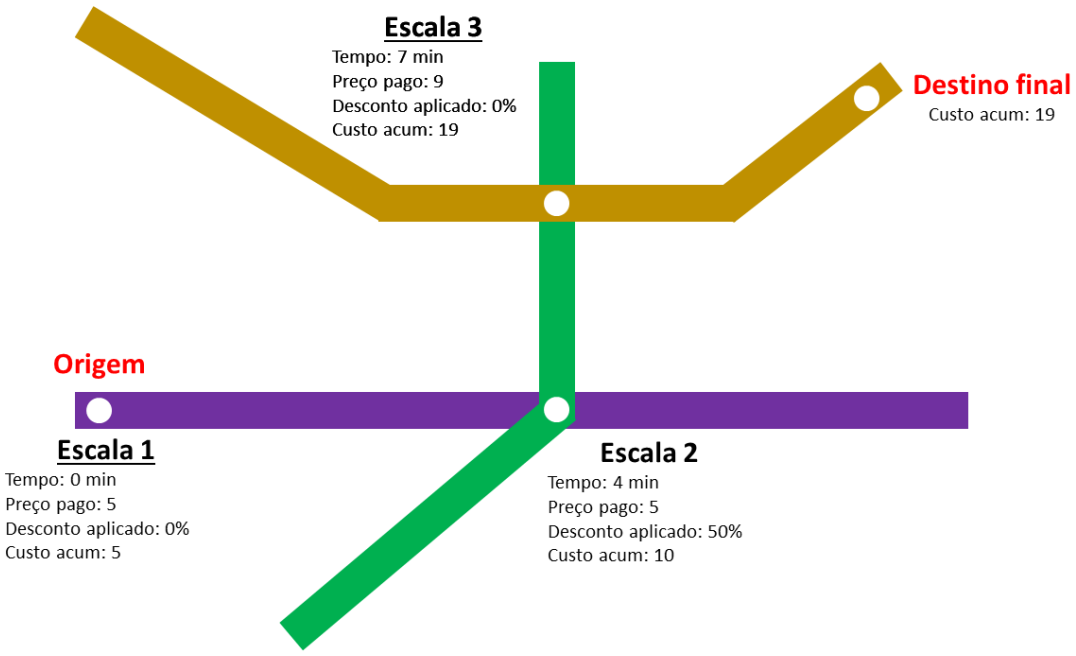

Figura 1: Diagrama da solução do exemplo 1 *(Escalas sublinhadas indicam início da contagem de tempo para aplicação de descontos)*

Exemplo de saída

```
19.00
```

Neste primeiro exemplo, conforme apresentado na Figura 1, aplica-se o desconto para a segunda viagem, de custo 10, diminuindo seu valor pela metade (5). Dessa forma, o custo final será: $5 + 5 + 9 = 19$.

4.2 Segundo exemplo

Exemplo das linhas da entrada

```
7 6 120
0 50 25 0 0 0
10 1
10 2
10 4
10 4
10 4
10 4
10 1
```

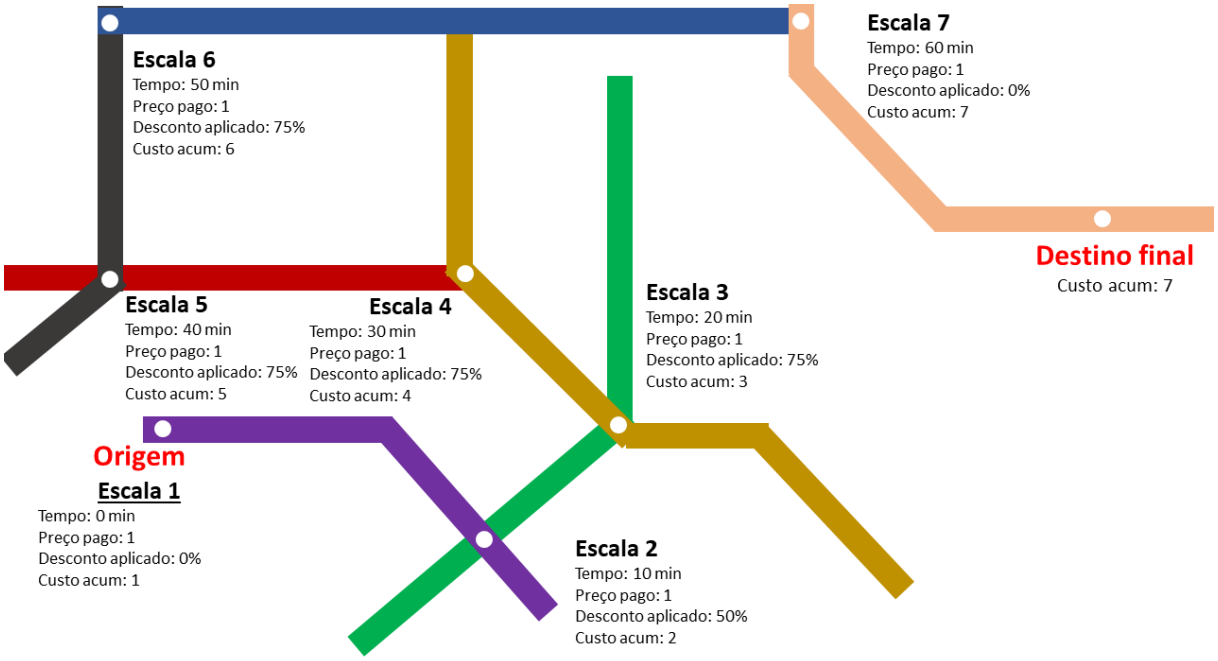

Figura 2: Diagrama da solução do exemplo 2 *(Escalas sublinhadas indicam início da contagem de tempo para aplicação de descontos)*

Exemplo linha de saída

```
7.00
```

Neste segundo exemplo, conforme apresentado na Figura 2, aplicam-se descontos entre a segunda e sexta escala, pois ao pegar a sexta escala transcorreram-se apenas 50 minutos (abaixo do prazo de 120 minutos). Na sétima escala, apesar de ter transcorrido apenas 60 minutos, não é mais aplicado desconto no bilhete, pagando-se o preço cheio.Dessa forma, o custo final será 7,00.

4.3 Terceiro exemplo

Exemplo das linhas da entrada

```
10 6 120
0 20 80 0 0 0
110 10
10 30
45 101
10 15
50 20
70 30
20 10
99 25
60 100
50 20
```

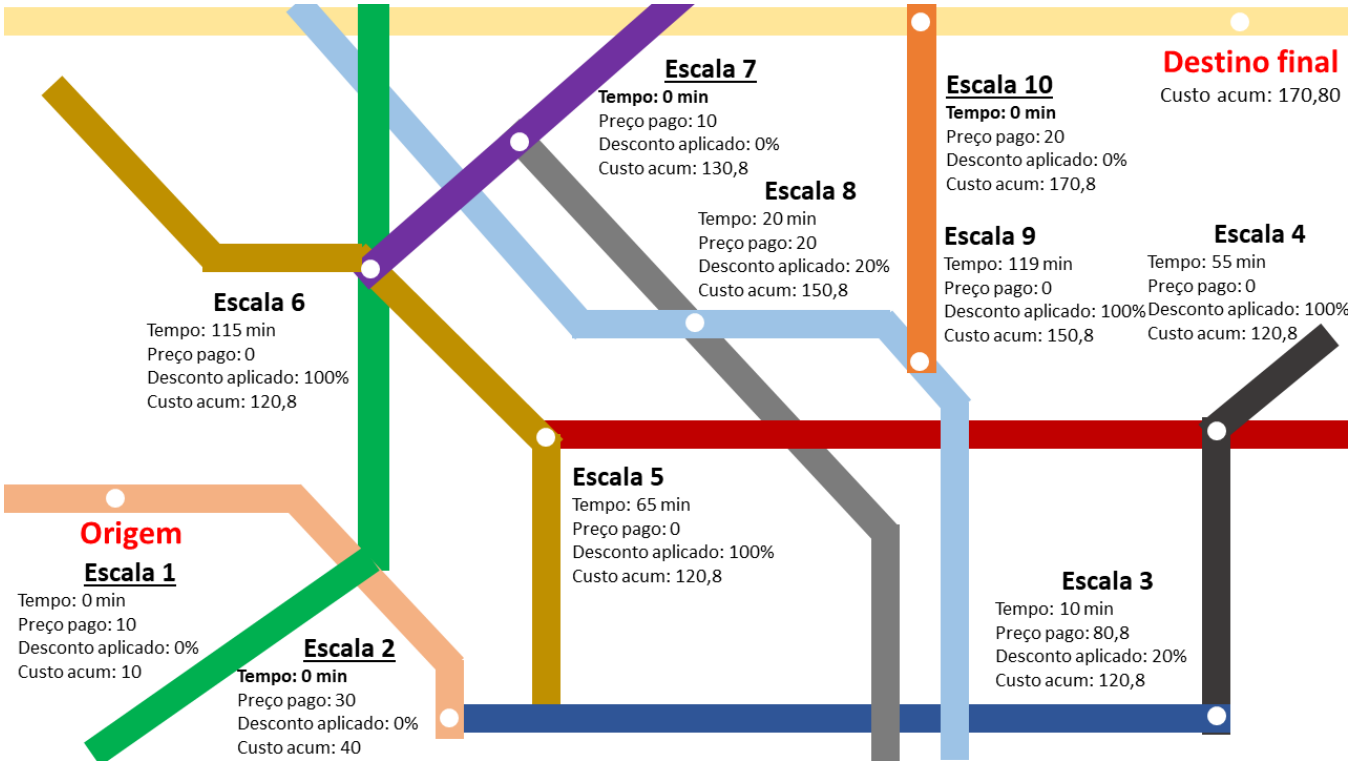

Figura 3: Diagrama da solução do exemplo 3 *(Escalas sublinhadas indicam início da contagem de tempo para aplicação de descontos)*

Exemplo das linhas da saída

```
170.80
```

Analisando a solução do terceiro exemplo pela Figura 3, observa-se que, como o tempo de translado da escala 1 é de 110 minutos, para João é mais interessante aguardar mais 10 minutos antes de iniciar a escala 2 para que finde o prazo de tempo dos descontos iniciados, do que prosseguir sem tempo de espera, situação na qual ele teria que pagar o valor cheio na escala 3 (de 101). Assim, João decide pagar o valor cheio do bilhete na escala 2, obtém 20 % de desconto na escala 3 e 100 % nas escalas 4,5 e 6.

Durante o translado da escala 6 para 7, João perde direito a desconto na próxima escala pois o tempo de 120 minutos é alcançado. Assim, ele paga o bilhete cheio na escala 7, obtém 20 % de desconto na escala 8, 100 % na escala 9.

Por fim, alcançado o tempo de direito de descontos iniciados na escala 7 durante o translado da escala 9, João paga o bilhete cheio na escala 10, obtendo o custo mínimo de 170,80 para chegar ao seu destino final.

4.4 Outros casos de teste

O aluno poderá realizar o *download* de outros casos de teste (bem como as respostas esperadas) para validação preliminar de seu programa no *moodle*.

5 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 tp03**;
  - **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 **tp03**.
- sua documentação (arquivo pdf).

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

- Modelagem computacional do problema, **incluindo a equação de recorrência no caso ter sido implementada programação dinâmica**;
- 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.

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

6 Implementação

6.1 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.

6.2 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.

6.3 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.

7 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.

7.1 Fator parcial de implementação

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

O fator parcial de implementação será calculado pela seguinte fórmula:

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

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

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

7.2 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.

7.3 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)