TP 01 - Trabalho Prático 02

Algoritmos I

Entrega: 02/08/2021

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 02/08/2021. A especificação do conteúdo do relatório e linguagens de programação aceitas serão detalhadas nas seções subsequentes.

2 Definição do problema

Neste trabalho prático você foi contratado por uma empresa aérea para auxiliar no gerenciamento de voos durante a pandemia. Esta companhia aérea realizava voos de ida e volta regularmente para diversos países do mundo. Entretanto, em razão da crise sanitária estabelecida pelo corona vírus e o surgimento de algumas variantes, alguns países decidiram proibir certos voos para esta companhia. Sendo assim, algumas rotas bidirecionais entre países ficaram unidirecionais ou deixaram de existir nas condições normais, e o mapa de voos resultante pode conter lacunas que não permitem voos entre alguns aeroportos, mesmo com escalas.

Entretanto, é facultada a criação de rotas adicionais seguindo protocolos rígidos de testagem, limitação de passageiros, utilização de máscaras específicas e outras medidas restritivas que culminam em custos adicionais elevados para a companhia. Ou seja, mediante um elevado custo para a companhia, é possível a criação de novas rotas.

Neste cenário, a companhia deseja saber qual o número mínimo de rotas que deverá adicionar à sua rede aérea para fazer com que um dado aeroporto de origem possa atingir qualquer aeroporto de destino. A rede aérea contém um conjunto de aeroportos $A = \{1, 2..., n\}$ operados pela companhia e um conjunto de rotas direcionadas $R = \{r_1, r_2, \ldots, r_m\}$, tal que $r_i = (u, v) \in A \times A$. Cada item de R é composto de um aeroporto de origem u e um aeroporto de destino v. Seu trabalho será encontrar o menor número de rotas adicionais $\{r_{m+1} \ldots\}$ que a companhia deve prover para que seus passageiros possam alcançar qualquer outro aeroporto. Note que poderá ser necessário para os passageiros fazer várias escalas até chegar no seu destino final.

3 O que fazer?

O objetivo do trabalho é identificar o número mínimo de rotas que precisam ser adicionadas em uma malha aérea para permitir que um determinado aeroporto de origem consiga atingir todos os aeroportos que são operados pela companhia aérea.

Serão fornecidos como entrada o número de aeroportos n operados pela companhia, o número de rotas m e o conjunto da rotas R que restaram após a imposição das restrições. Seu programa deverá identificar o número mínimo de rotas que devem ser adicionadas à rede para que um passageiro possa chegar a qualquer aeroporto de destino a partir de qualquer aeroporto de origem.

4 Exemplo do problema

Um exemplo do problema é apresentado na figura 1, onde o objetivo é encontrar o mínimo de rotas que devem ser acrescentadas para que se possa voar de um aeroporto para qualquer outro. Trata-se de uma companhia aérea que opera em 15 aeroportos (1 a 15) e possui 18 rotas direcionadas (na representação gráfica, as arestas bidirecionais devem ser entendidas como duas rotas, uma de ida e outra de volta).

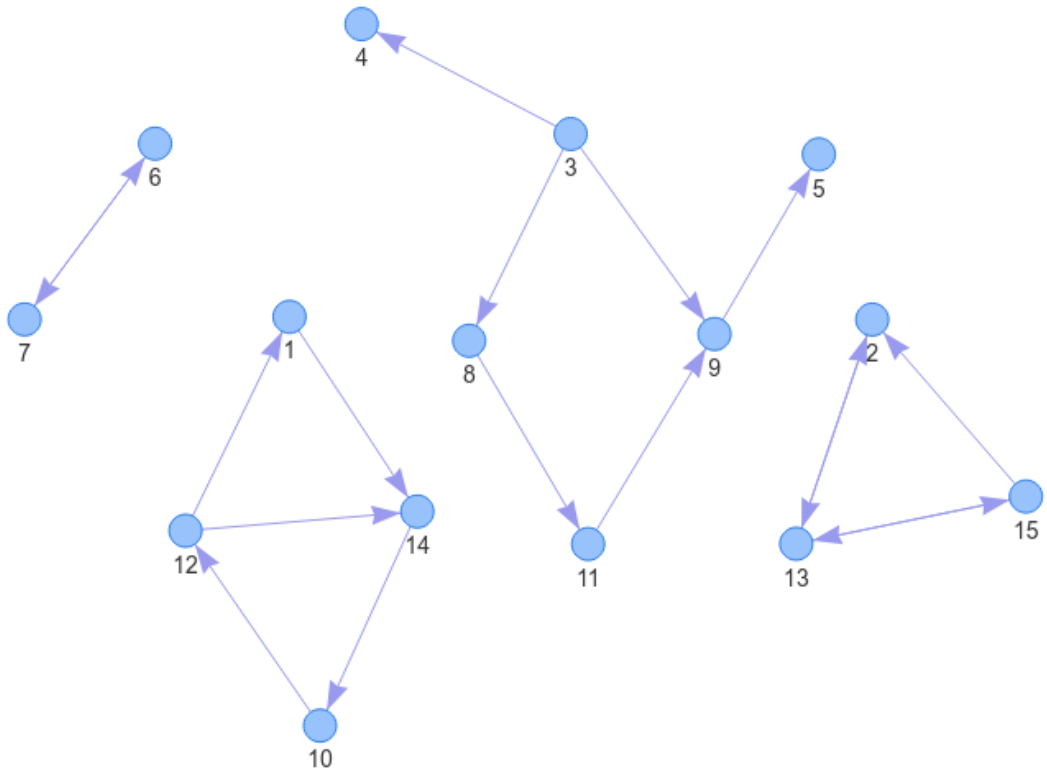

Figura 1: Rotas da companhia aérea

Com esta configuração, para que a companhia possa atingir seu objetivo de permitir viagens entre quaisquer aeroportos, o número mínimo de rotas adicionais deverá ser igual a 5. A figura 2 ilustra um arranjo possível de rotas (em vermelho) que atendem ao objetivo dessa configuração. Observe que não é necessário especificar quais são as rotas adicionais. Ou seja, a resposta esperada do programa é apenas o número mínimo de rotas adicionais.

Os dados do problema ilustrado na figura 1 são exibidos a seguir.

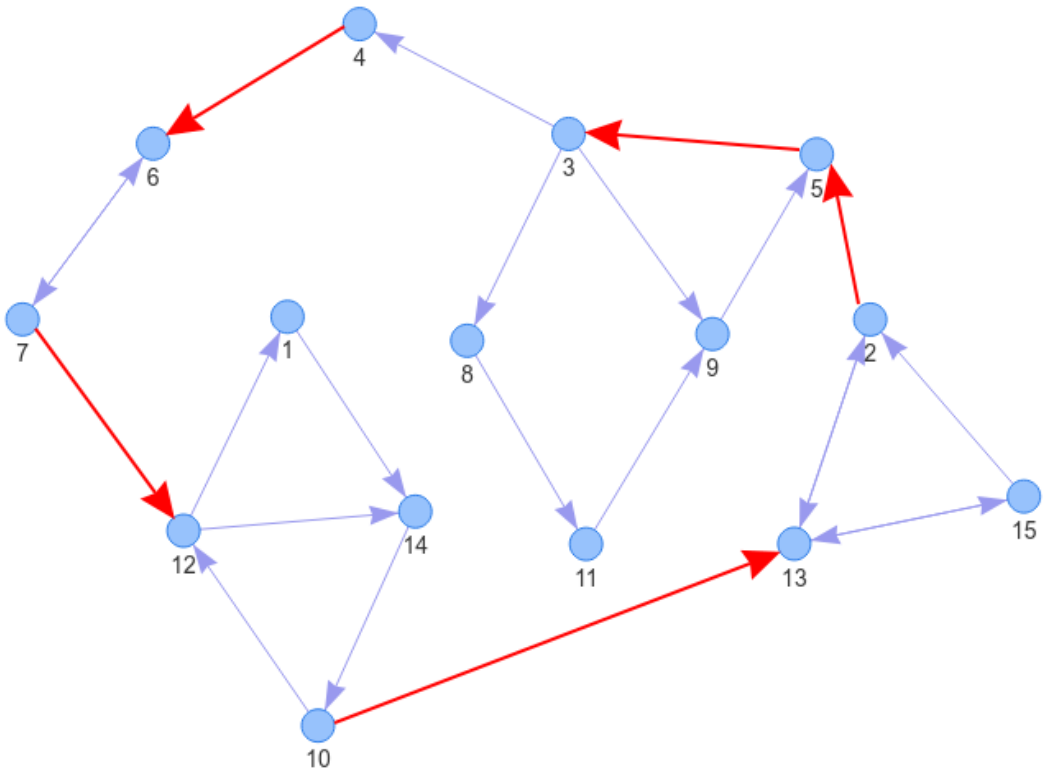

Figura 2: Rotas da companhia aérea com acréscimos

Dados de entrada da figura 1.

```
15 18                                   número_de_aeroportos número_de_rotas
1 14                                    lista de rotas
2 13
3 4
3 8
3 9
6 7
7 6
8 11
9 5
10 12
11 9
12 1
12 14
13 2
13 15
14 10
15 2
15 13
```

Um segundo exemplo é exibido na Figura 3, contendo 50 aeroportos e 65 rotas. O número mínimo de rotas adicionais para conectar todos aeroportos é 7 (essa é a resposta esperada) e uma solução possível é exibida na Figura 4.

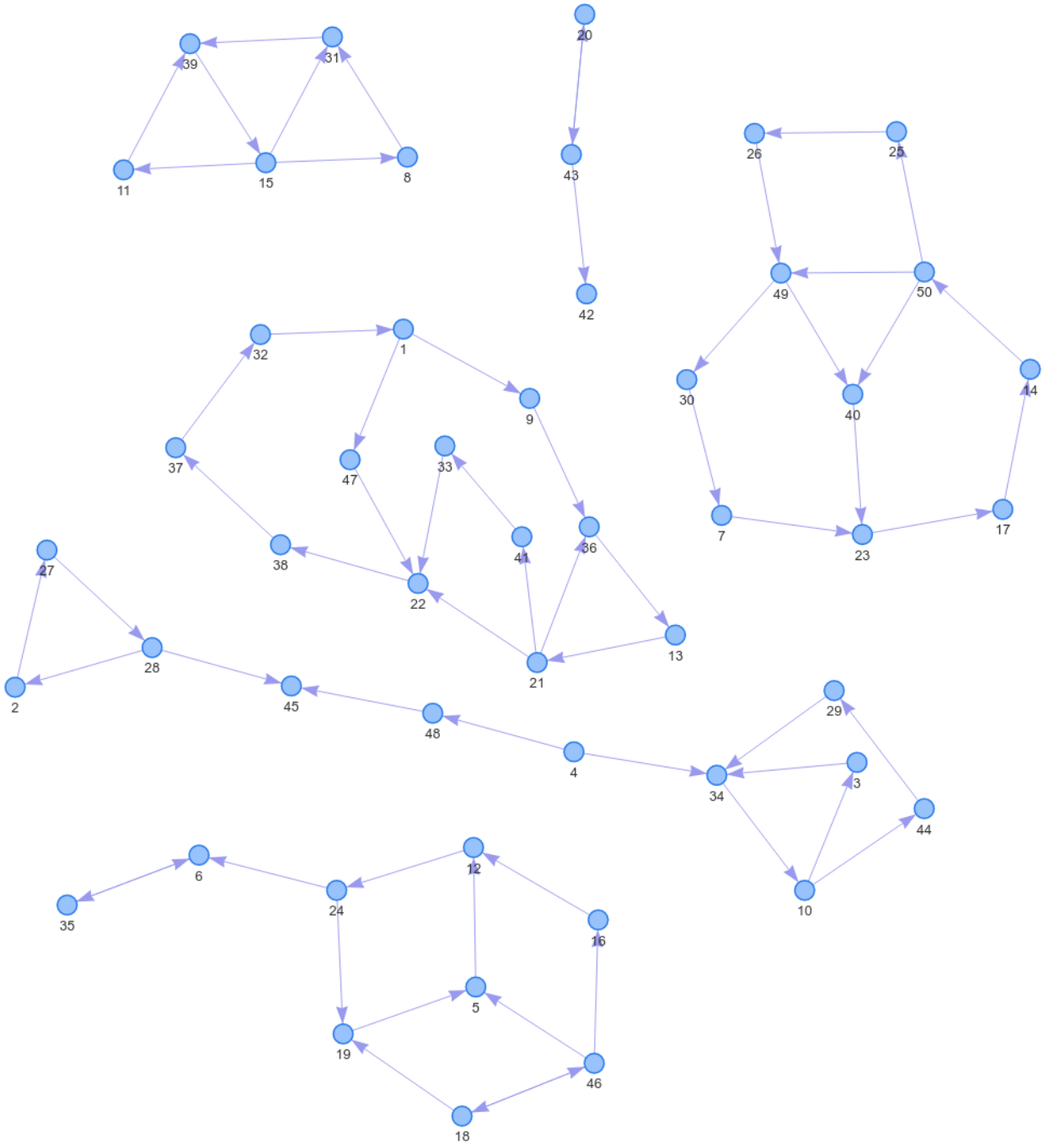

Figura 3: Rotas da companhia aérea

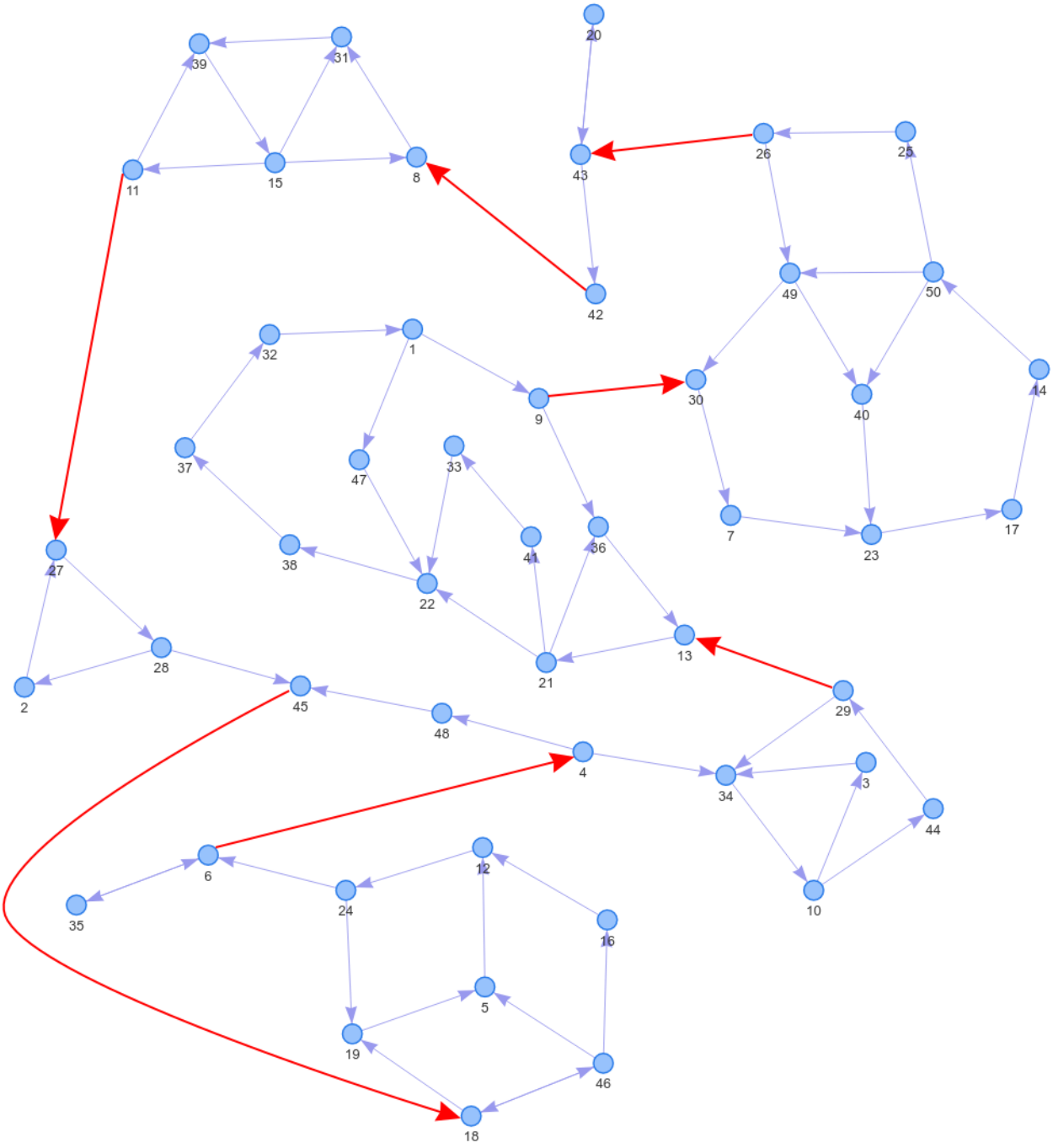

Figura 4: Rotas da companhia aérea com acréscimos

5 Arquivo de entrada

O arquivo de entrada segue o padrão do exemplo fornecido para a figura 1. A primeira linha contém o número de aeroportos n, (indicando a existência dos aeroportos 1, 2, ... n) e o número de rotas m, separados por um espaço, com $0 \leq n \leq 10^4$ e $0 \leq m \leq 10^4$. As demais linhas contêm as rotas partindo de um aeroporto de origem u para um aeroporto de destino v, também separados por espaço.

6 Saída

A saída do programa deverá ser uma única linha (não esqueça de imprimir "\n" no final da linha) contendo o número mínimo de rotas que devem ser adicionadas à rede aérea da companhia para que um vôo de qualquer um dos aeroportos possa chegar a qualquer outro aeroporto operado pela companhia, independente do número de escalas.

7 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 tp02**;
  - **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 **tp02**.
- 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;
- estruturas de dados e algoritmos utilizados para resolver o problema (pseudo-código da solução implementada), bem como justificativa para tal escolha. Não transcreva trechos da código-fonte;
- análise de complexidade de tempo assintótica da solução proposta, devidamente justificada.

8 Implementação

8.1 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., *$ ./tp02 < casoTeste01.txt*) e gerar o resultado também na saída padrão (não gerar saída em arquivo).

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

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.

8.2 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.

8.3 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.

9 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.

9.1 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 de 5s	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.

9.2 Fator parcial da documentação

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

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.

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

9.3 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)