TP1 - O Plano de Campanha

Algoritmos 1
Data de entrega: 11/10/2022

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. A sua solução deve obrigatoriamente ser desenvolvida utilizando algoritmos de grafos. 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 11/10/2022. A especificação do conteúdo do relatório e linguagens de programação aceitas serão detalhadas nas seções subsequentes.

2 Definição do problema

O atual deputado federal Alan Turing está tentando se reeleger, então escreveu um plano de campanha com várias propostas diferentes para abranger a maior quantidade de eleitores possível. Infelizmente, com tantas propostas, o candidato acabou se contradizendo e por isso ele precisa repensar seu plano de campanha.

Por sugestão de seu assessor, o deputado fez uma enquete no seu Instagram para que cada seguidor fizesse 4 escolhas. Primeiro o seguidor deve escolher duas propostas que ele acha que devem continuar no projeto de campanha, e em seguida escolher duas propostas que ele acha que devem ser retiradas do plano de campanha.

Após a votação o deputado **deseja saber se existe um conjunto de propostas que agrade todos os seus seguidores simultaneamente**. Como esta é uma tarefa muito difícil, foi decidido que **para agradar um seguidor ao menos uma das propostas votadas a favor deve continuar no plano de campanha do deputado, e ao menos uma das propostas com voto contrário deve ser retirada do plano**.

3 Exemplo do Problema

O deputado possui 3 seguidores no seu Instagram e publicou 4 propostas diferentes para serem votadas. O primeiro seguidor votou a favor das propostas 1 e 2 e votou contra as propostas 3 e 4. O segundo seguidor votou a favor das propostas 3 e 4 e votou contra as propostas 1 e 2. E o último seguidor votou a favor das propostas 2 e 3 e votou contra as propostas 1 e 4. Para essa rodada de votações, é possível que o deputado agrade a todos os seguidores que votaram quando escolhemos manter as propostas 2 e 3 e removemos as propostas 1 e 4. Podemos visualizar isso melhor na figura 1.

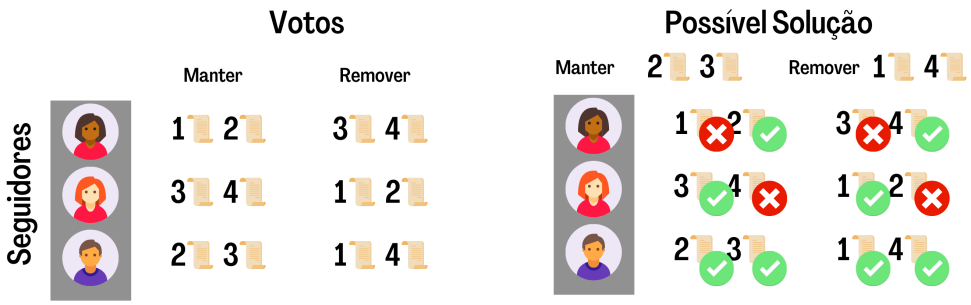

Figure 1: Exemplo com possível solução. Nela podemos ver os votos dos seguidores e quais votos foram satisfeitos ou não. Observe que pelo menos uma escolha de manter de cada seguidor foi respeitada, assim como pelo menos uma escolha de retirada de cada seguidor foi respeitada.

3.1 Modelagem do Problema

Este trabalho prático aborda a parte de grafos da ementa desta disciplina. Para a resolução do problema a sua modelagem **precisa** usar grafos.

3.2 Formato da Entrada Esperada

O seu programa deverá processar diversos casos de teste em uma única execução. A primeira linha de um cenário de teste é composta de dois número inteiros S e P, representando respectivamente o número de seguidores que responderam a enquete ($1 \leq S \leq 1000$) e o número de propostas do plano de campanha ($1 \leq P \leq 10000$). Cada uma das próximas S linhas descreve a preferência de um seguidor, representada por quatro inteiros X_1, X_2, Y_1, Y_2 ($0 \leq X_1, X_2, Y_1, Y_2 \leq P$). X_1 e X_2 são propostas que o seguidor deseja manter no plano de campanha. Y_1 e Y_2 são propostas que o seguidor deseja retirar do plano de campanha. Um valor 0 (zero) para qualquer uma das variáveis X_1, X_2, Y_1 e Y_2 significa que o seguidor não está fazendo uso daquele voto. A entrada acaba quando $S = P = 0$.

3.3 Formato da Saída Esperada

Para cada caso de teste seu programa deve imprimir uma linha, contendo ou a palavra 'sim' (se é possível satisfazer a todos os seguidores que votaram) ou a palavra 'nao' (se não é possível).

3.4 Caso de teste

Entrada	Saída
3 4 0 2 3 4 0 4 1 2 2 3 1 4 4 4 1 2 3 4 3 4 1 2 1 3 2 4 1 4 2 3 4 4 1 2 3 4 3 4 1 0 1 3 2 4 2 4 0 3 7 6 1 3 4 2 1 2 4 5 2 3 5 1 4 5 1 6 1 2 3 4 1 4 3 6 2 6 1 3 0 0	sim sim nao nao .

4 Implementação

O seu programa deverá ser implementado na linguagem 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 ou login.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/).

Para facilitar o desenvolvimento vamos fornecer uma estrutura base de arquivos com Makefile já configurado. A pasta `TP01-Template-CPP.zip`, disponível para download na tarefa do Moodle, contem 4 arquivos: `main.cpp`, `campanha.cpp`, `campanha.hpp` e `Makefile`. O ponto de entrada do seu programa está no arquivo `main.cpp`. Para compilar seu programa basta executar o comando "`make`" no mesmo diretório que o Makefile está. Ao final deste comando, se a compilação for bem sucedida, será criado um arquivo executável chamado "`tp01`". Esse arquivo pode ser executado pela linha de comando usando "`./tp01`".

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).

5 O que deve ser entregue

Deverá ser submetido um arquivo .zip contendo apenas uma pasta chamada tp1, esta pasta deverá conter: (i) Documentação em formato PDF e (ii) Implementação.

5.1 Documentação

A documentação deve ser sucinta e não ultrapassar 5 páginas. Você deve descrever cada solução do problema de maneira clara e precisa, detalhando e justificando os algoritmos e estruturas de dados utilizados. Para tal, artifícios como pseudo-códigos, exemplos ou diagramas podem ser úteis. Note que documentar uma solução não é o mesmo que documentar seu código. Não é necessário incluir trechos de código em sua documentação nem mostrar detalhes de sua implementação, exceto quando estes influenciem o seu algoritmo principal, o que se torna interessante. É essencial que a documentação contenha ao menos:

1. **Identificação**: Nome e Matrícula.
2. **Introdução**: Breve resumo do problema com suas palavras.
3. **Modelagem**: Detalhamento e justificativa dos algoritmos e estruturas de dados escolhidos.

5.2 Implementação

O código fonte submetido deve conter todos os arquivos fonte e o Makefile usado para compilar o projeto. Lembre que seu código deve ser **legível**, então **evite variáveis com nomes não descritivos** (`int a, aa, aaa;`) e lembre-se de **comentar seu código**. Já estamos fornecendo uma implementação base com os arquivos necessários, então indicamos que você só o altere se for necessário.

5.3 Atrasos

Trabalhos poderão ser entregues após o prazo estabelecido, porém sujeitos a uma penalização regida pela seguinte fórmula:

$$\Delta_p = \frac{2^{d-1}}{0.32}\% \tag{1}$$

Nesta fórmula d representa dias de atraso. Por exemplo, se a nota dada pelo corretor for 70 e você entregou o TP com 4 dias corridos de atraso, sua penalização será de $\Delta_p = 25\%$ e, portanto, a sua nota final será: Nf$= 70(1 - \Delta_p) = 52.2$. Note que a penalização é exponencial e 6 dias de atraso resultam em uma penalização de 100%.

6 Considerações Finais

Assim como em todos os trabalhos dessa disciplina é estritamente proibida a cópia parcial ou integral de códigos, seja da internet ou de colegas. Utilizaremos o algoritmo MOSS para detecção de plágio em trabalhos, seja honesto. Você não aprende nada copiando código de terceiros nem pedindo a outra pessoa que faça o trabalho por você. Se a cópia for detectada, sua nota será zerada e os professores serão informados para que as devidas providências sejam tomadas.