TP3 - Exposição de tecidos

Algoritmos 1
Data de entrega: 11/12/2022

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 programação dinâmica.

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 **11/12/2022**. A especificação do conteúdo do relatório e linguagens de programação aceitas serão detalhadas nas seções subsequentes.

2 Definição do problema

Jorginho é um exímio vendedor na loja de tecidos FP Tecidinhos. Ao início de cada dia, ele recebe uma remessa de rolos de tecido. Cada rolo é consideravelmente pesado, por isso Jorginho só pode manipular os rolos na ordem em que eles chegaram. Para posicionar cada rolo, o vendedor tem três opções: **1) colocá-lo na prateleira pelo lado direito e empurrá-lo até encostar nos rolos já existentes, 2) colocá-lo na prateleira pelo lado esquerdo e empurrá-lo até encostar nos rolos já existentes, ou 3) simplesmente não colocá-lo na prateleira**. Se não existir nenhum rolo na prateleira, Jorginho pode colocá-lo de qualquer direção que não faz diferença. Além da restrição de como colocar os produtos, Jorginho precisa que os **rolos estejam em ordem decrescente de preço na prateleira**. Ele aprendeu que assim consegue vende-los mais facilmente. Vale lembrar também que como os rolos são muito grandes eles são todos de tecidos diferentes, logo, Jorginho **nunca encomenda produtos de valores iguais.**

Dado uma remessa de rolos, em que se conhece a ordem e o preço dos produtos, **qual é a maior quantidade de rolos que Jorginho consegue expor** seguindo as restrições acima ?

3 Exemplo do Problema

Em um caso hipotético, chegaram 4 rolos para Jorginho organizar na prateleira.

- O primeiro rolo tem valor de 6 reais.
- O segundo rolo tem valor de 7 reais.
- O terceiro rolo tem valor de 3 reais.
- O quarto rolo tem valor de 5 reais.

Seguindo a forma de organização de Jorginho, a prateleira da loja terá 3 rolos de tecidos expostos. Observe os passos tomados na figura 1.

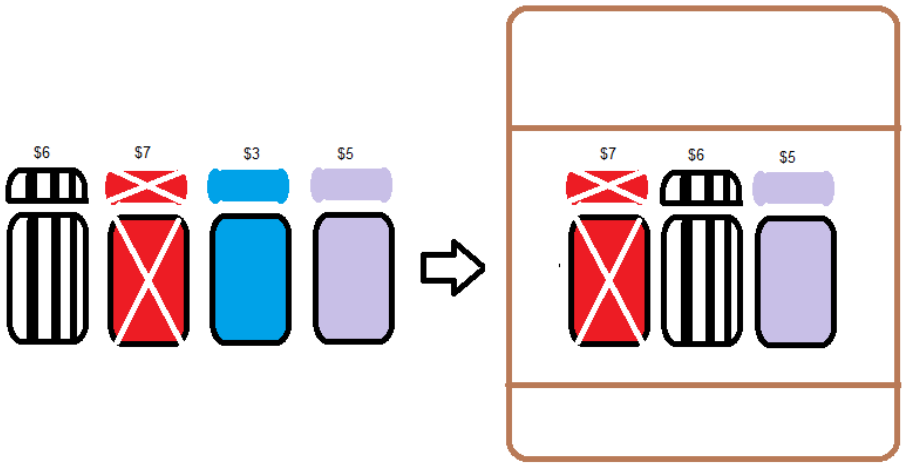

Figure 1: Exemplo de como 4 rolos de tecidos serão expostos na prateleira. Primeiro Jorginho inseriu o rolo de custo 6 na prateleira. Em seguida inseriu o rolo de custo 7 à esquerda da prateleira, depois decidiu não inserir o rolo de custo 3, e inseriu o rolo de custo 5 à direita da prateleira. Note que ao final, os preços dos produtos estão em ordem decrescente.

3.1 Modelagem do Problema

Este trabalho prático aborda a parte de programação dinâmica a da ementa desta disciplina. Para a resolução do problema a sua modelagem **precisa** usar um algoritmo de programação dinâmica com complexidade $O(n^2)$. **É obrigatório que a sua documentação contenha a relação de recorrência e a análise de complexidade do seu algoritmo**.

3.2 Formato da Entrada Esperada

O seu programa deverá processar diversos casos de teste em uma única execução. A primeira linha contém um inteiro N, tal que $1 \leq N \leq 10$, que representa a quantidade de casos de teste. Em seguida, para cada teste a próxima linha contém um inteiro R, tal que $1 \leq R \leq 20000$, que representa a quantidade de rolos que chegarão a loja. Cada uma das próximas R linhas contém um inteiro P_i, tal que $1 \leq P_i \leq R$, que é o valor do i-ésimo rolo de tecido.

3.3 Formato da Saída Esperada

Para cada caso de teste seu programa deve imprimir uma linha contendo um número inteiro que significa a quantidade de rolos que serão expostos.

3.4 Caso de teste

Entrada

```
3
4
6
7
3
5
9
9
1
3
8
4
5
10
7
6
10
3
13
1
4
9
8
12
6
14
5
```

Saída

```
3
5
6
```

.

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 `TP03-Template-CPP.zip`, disponível para download na tarefa do Moodle, contem 4 arquivos: `main.cpp`, `tecido.cpp`, `tecido.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 "`TP03`". Esse arquivo pode ser executado pela linha de comando usando "`./tp03`".

O arquivo da entrada deve ser passado ao seu programa como entrada padrão, através da linha de comando (e.g., `$ ./tp03 < casoTeste03.txt`) e gerar o resultado também na saída padrão (não gerar saída em arquivo).

Por favor não altere o nome da pasta nem o nome do arquivo executável. Parte da correção é automática e é extremamente importante manter o padrão de execução.

5 O que deve ser entregue

Deverá ser submetido um arquivo .zip contendo apenas uma pasta chamada tp3, 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. Inclui a relação de recorrência e a análise de complexidade.

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.