

Universidade Tecnológica Federal do Paraná – Campus Apucarana
Estruturas de Dados (ED62A) – 9º semestre
Prof. Dr. Rafael Gomes Mantovani

Instruções:

- A interpretação dos enunciados faz parte da avaliação;
- É permitido o uso de rascunho no auxílio das questões. Contudo, os rascunhos devem ser identificados e entregues junto com a prova;
- É permitido consultar a folha com os tipos definidos e as assinaturas dos métodos que as manipulam.
- As respostas das questões que envolvem o desenvolvimento de código devem ser elaboradas em C. Contudo, soluções em pseudocódigo podem ser avaliadas em até metade da nota do exercício;

Exercício 1. (2,00 pontos) Na resolução de um problema por meio de um programa, a primeira providência é conceber um algoritmo adequado. A eficiência de um algoritmo está intimamente relacionada à disposição, na memória, dos dados que são tratados por um programa. Desta forma, faça um comparativo entre os diferentes tipos de estruturas de dados elementares. Lembre-se também de analisá-las levando em conta o tipo de implementação.

Exercício 2. (Pilhas - 2,00 pontos) O Desafio das Torres de Hanoi (Figura 1) é um jogo clássico, inspirado em uma antiga lenda. O jogo conta com três hastes verticais (A, B, C). Uma das hastes contém vários discos de diâmetros diferentes (1 a 5). O desafio do jogo é passar todos os discos, nessa mesma sequência para uma das outras duas hastes. É possível utilizar as três hastes para auxiliar a movimentação dos discos. Mas existem algumas restrições na movimentação:

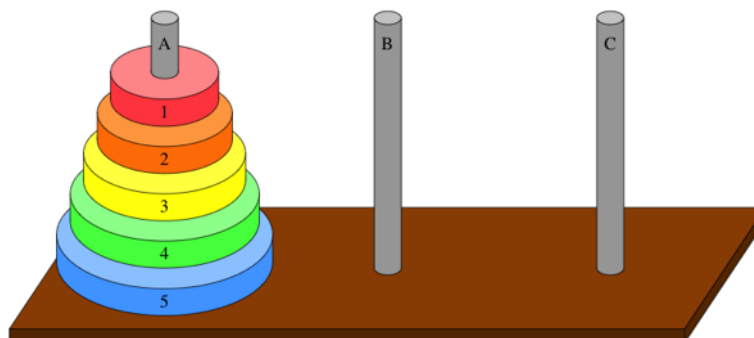


Figura 1: Exemplo de Torres de Hanoi com 5 discos

- só é possível movimentar um disco de cada vez;
- só é possível inserir discos no topo de uma pilha de discos;
- em nenhum momento você pode colocar um disco em cima de outro disco com diâmetro menor.

Considerando que cada uma das hastes pode ser abstraída como uma **Pilha**, defina uma sequência de movimentos (push, pop) entres Pilhas para mover os todos os discos da haste A para uma das outras duas hastes. Desenhe/represente os estados intermediários das pilhas após as movimentações. **Cuidado:** remover o disco 1 e inserir na Pilha B envolve duas operações $\rightarrow x = \text{pop}(A)$ e $\text{push}(B, x)$.

Exercício 3. (Filas - 2,00 Pontos) Considere uma área de armazenamento de dados em um computador com **3 espaços** físicos. Sobre esta área está montada uma fila de nome **FILA**. Qual o comportamento da fila durante as seguintes operações descritas na tabela 1 ? Desenhe e identifique o estado da fila **em cada** uma das operações.

Tabela 1: Sequência de operações em um tipo Fila.

Id	Operação
1	Inserção de um nó com valor VERMELHO
2	Inserção de um nó com valor VERDE
3	Inserção de um nó com valor AMARELO
4	Inserção de um nó com valor BRANCO
5	Inserção de um nó com valor PRETO
6	Remoção de um nó
7	Remoção de um nó
8	Inserção de um nó com valor ROSA
9	Remoção de um nó
10	Remoção de um nó
11	Inserção de um nó com valor AZUL
12	Inserção de um nó com valor CINZA
13	Remoção de um nó
14	Remoção de um nó

Exercício 4. (Filas - 2,00 ponto) Desenvolva um algoritmo para trocar elementos de duas Filas F1 e F2. Ao final da operação, a fila F2 deve conter os elementos que estavam em F1 antes do início da operação, na mesma ordem, e a fila F1 deve conter os elementos que estavam em F2 antes do início da operação, também na mesma ordem. Considere que F1 e F2 já existem e são passadas como parâmetros.

Exercício 5. (Listas - 2,00 Pontos) Implemente uma função que receba um vetor de valores inteiros com n elementos e construa uma lista encadeada armazenando os elementos do vetor v nos nós da lista. Se o vetor tiver zero elementos, a função deve ter como valor de retorno uma lista vazia. O protótipo da função é dado por:

```
Lista* constroi(int n, int* v);
```

Exercício 6. (Pilhas - 3,00 Pontos) O *FreeCell*¹ é um jogo de paciência, que pode ser jogado também com um baralho de verdade e não apenas no mundo virtual. O *FreeCell* ficou muito conhecido depois que passou a ser distribuído junto com o sistema operacional Windows (Figura 2). Inicialmente, as 52 cartas de um baralho são distribuídas aleatoriamente em pilhas de cartas, fique ficam na parte de baixo da tela. Essas pilhas são chamadas de *Pilhas intermediárias*. O objetivo é empilhar todas as 52 cartas nas pilhas que ficam no canto superior direito da tela, chamadas de *Pilhas Definitivas*. Mas, nas *Pilhas Definitivas* é preciso empilhar as cartas em sequência (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K) e organizadas por naipes (copas, espadas, paus e ouros). Há uma *Pilha Definitiva* para cada naipe.

A movimentação das cartas precisa seguir algumas regras:

- Só é possível retirar a carta que está no topo de uma *Pilha Intermediária*;
- É possível passar cartas de uma *Pilha Intermediária* para outra, mas inserindo as cartas sempre no topo da pilha, em ordem decrescente, e em cores alternadas - pretas sobre vermelhas, e vermelhas sobre pretas.

Existem ainda quatro *Espaços para Movimentação de Cartas*, no canto superior esquerdo da tela. Nesses espaços é possível colocar qualquer carta, temporariamente. O uso desses espaços pode ajudar bastante no andamento do jogo.

Suponhamos que você tenha sido contratado para implementar esse jogo. Como você começaria o trabalho? Pense em todas as escolhas na modelagem do problema.

- a) **(1,50 pontos)** Quais estruturas seriam necessárias para armazenar os valores dos naipes e cartas? Qual seria a melhor estratégia para armazenar e manipular as informações referentes a essas pilhas

¹<https://freecell.eco.br/paciencia-freecell/>



Figura 2: Tela de jogo do *FreeCell*.

de cartas dentro do seu programa? Defina o(s) tipo(s) abstrato(s) de dados para que se possa implementar e manipular suas diferentes pilhas dentro do jogo, e especifique quais tipos seriam usados para implementar quais pilhas (intermediárias e definitivas).

- b) **(1,50 pontos)** Além disso, como você armazenaria a sequência de cartas em cada uma das pilhas (definitivas e intermediárias) e implementaria as regras que restringem a movimentação das cartas? Nesse sentido, defina duas funções 'inteligentes': uma para realizar a inserção de cartas em uma *Pilha Intermediária*, e outra para inserir cartas em uma *Pilha Definitiva*.

Dicas: Nas pilhas definitivas só podem ser armazenadas cartas de um mesmo naipe, e valores inseridos de forma crescente. Já nas pilhas intermediárias, as inserções são feitas estabelecendo a ordem decrescente dos símbolos das cartas, e alternando as cores dos naipes (preta sobre vermelha, e vermelha sobre preta). Se necessário, acesse o jogo pelo link no rodapé da página anterior, e jogue algumas partidas para entender a lógica e regras do jogo.