MAC–IME–USP

Alfredo Goldman
Sala B122
e-mail gold@ime.usp.br

MAC 2301 – Laboratório de Estruturas de Dados

Primeiro semestre de 2001

Exercício-Programa 2 – Date de Entrega: **2 de maio de 2001**

Jogo de paciência

1 Introdução

O objetivo desse exercício–programa é simular um jogo de paciência que descrevemos a seguir. Esse jogo é jogado com apenas 1 baralho (52 cartas). Distribuem-se 28 dessas cartas em 7 pilhas de jogo da seguinte forma: coloque uma carta em cada pilha de jogo da esquerda para a direita, tal que na i-ésima passada você começa na i-ésima pilha colocando uma carta aberta e todas as outras cartas ficam fechadas. As demais cartas ficam na chamada *pilha de estoque*. Ainda existe uma *pilha de descarte* e 4 *pilhas de saída*. O objetivo do jogo é conseguir o maior número possível de cartas nas pilhas de saída.

2 Regras

Nas pilhas de jogo você só pode colocar cartas que formem uma seqüência decrescente consecutiva e de cores alternadas. Assim, se o topo de uma pilha de jogo é um 9 de espadas, você só poderá colocar sobre ele um 8 de ouros ou um 8 de copas. Todas as cartas abertas de uma pilha de jogo podem ser removidas como uma unidade e colocadas em outra pilha de jogo, desde que se respeite a regra de a seqüência ser decrescente e de cores alternadas. Por exemplo, a seqüência 8 de copas, 7 de paus e 6 de ouros pode ser movida como uma unidade e colocada na pilha de jogo cujo topo é um 9 de paus.

Cada pilha de jogo deve ter sempre pelo menos uma carta aberta. Assim, se as cartas abertas forem removidas, a carta do topo deve ser aberta. Se uma pilha de jogo ficar vazia, ela só poderá ser reaberta por um Rei que esteja aberto em alguma pilha de jogo ou no topo da pilha de descarte.

Sempre que um ás aparece no topo de uma pilha de jogo ou na pilha de descarte ele é transferido para uma das pilhas de saída. Cada pilha de saída conterá as cartas de um naipe consecutivamente em ordem crescente. Não é permitido remover as cartas das pilhas de saída.

As cartas da pilha de estoque só podem ser removidas abertas para a pilha de descarte, uma de cada vez. Sempre que a pilha de descarte ficar vazia uma nova carta da pilha de estoque deve ser movida para lá.

Observe que pode ocorrer que diversos movimentos são possíveis. Nesse caso, você deve desenvolver uma estratégia que lhe pareça trazer melhores chances de ganhar o jogo. O jogo termina quando nenhum movimento for possível e a pilha de estoque estiver vazia.

Faça um programa que simule uma máquina de jogar paciência como descrito acima, permitindo que diversos jogos possam ser executados. Você deve permitir que o usuário entre com um baralho já embaralhado (que será um arquivo com 52 linhas, em que cada uma contém uma carta), ou deverá embaralhar aleatoriamente o baralho. Para vocês que não conhecem jogos, as cartas de um baralho estão divididas em 4 naipes (ouros O, copas C– que são vermelhos – paus P e espadas E– que são pretos). Para cada naipe existem 13 cartas: ás, 2, 3, 4, 5, 6, 7, 8, 9, 10, J (valete), Q (dama) e K (rei). Escolha uma implementação mais eficiente possível para armazenar as cartas. O formato do arquivo de entrada será dado por 52 linhas. Se a primeira linha for `10 O`, a primeira carta será um 10 de ouros. Se a segunda linha é `K P`, a segunda carta será um rei de paus, e assim por diante. A última carta do arquivo estará no topo do monte, e portanto será a primeira a ser distribuída.

Você deve entregar um disquete contendo o programa fonte e uma listagem do programa, ou enviar o código fonte diretamente ao seu monitor. A saída do seu programa deve ser clara. Obviamente seu programa deverá ser bem documentado, e a forma de utilizá-lo também será avaliada.

3 Implementação

Todas as pilhas[1] do programa devem ser representadas utilizando-se alocação ligada. Você também deve implementar uma lista livre (conforme visto em aula).

O seu programa também deve ser capaz de jogar automaticamente 100 partidas (geradas aleatoriamente) e imprimir a porcentagem de partidas onde foi possível colocar todas as cartas na pilha de saída.

Em caso de dúvidas, lembrem-se a lista de discussão pode ajudar.

4 Bônus

Você poderá ganhar alguns bônus em sua nota neste EP. Qualquer coisa que você implemente além do pedido no enunciado dá direito a um bônus. Documente isso no seu programa. Aí vão algumas idéias:

- Faça um programa que permita também ao usuário jogar;
- Verifique a diferença de desempenho quando não se usa lista livre (cada alocação e liberação de nós acessa diretamente a memória).
- Implemente uma versão que utilize um único vetor (primeiro com 70 e depois com 150 posições), discuta sobre a eficácia das duas versões (note que você deverá comentar as diferenças conforme o tamanho do vetor);
- Implemente versões diferentes do jogo (como nas implementações disponíveis no Windows).

[1] Implemente as pilhas conforme visto em aula, isto é, as funções empilha e desempilha devem receber como parâmetro cartas, e não nós.