MAC-2301 — Estruturas de Dados

Primeira lista de exercícios

1. Considere um estacionamento de trens (analogia com a pilha), como visto em aula.

   (a) Suponha que existam seis trens na entrada, numerados por $1, 2, 3, 4, 5$ e 6. Os trens poderão sair na ordem 325641? E na ordem 154623? (Justifique).

   (b) Escreva um algoritmo que dado $n > 0$ e dada uma permutação de $1, 2, \ldots, n$, decide se os trens poderão sair na ordem especificada na permutação dada.

2. Considere o problema do exercício anterior substituindo a pilha por uma fila dupla.

   (a) Determine uma permutação de $1, 2, 3$ e 4 que possa ser obtida com uma fila dupla com entrada restrita, mas que não possa ser obtida com uma fila dupla com saída restrita.

   (b) Determine uma permutação de $1, 2, 3$ e 4 que possa ser obtida com uma fila dupla com saída restrita, mas que não possa ser obtida com uma fila dupla com entrada restrita.

   (c) Determine uma permutação de $1, 2, 3$ e 4 que não possa ser obtida com uma fila dupla com entrada restrita e nem com uma fila dupla com saída restrita.

   Justifique cada um dos casos.

3. Reescreva a função abaixo utilizando uma **pilha** para eliminar, isto é, simular a recursão:

   (a)
```
void Bool(int i,int b1,int b2,int b3){
      if (i==1) {
        Bool(2,0,0,0); Bool(2,1,0,0); Bool(2,2,0,0);
      }
      else if (i==2) {
        Bool(3,b1,0,0); Bool(3,b1,1,0); Bool(3,b1,2,0);
      }
      else if (i==3) {
        Bool(4,b1,b2,0); Bool(4,b1,b2,1); Bool(4,b1,b2,2);
      }
      else
        printf("%d%d%d\n",b1,b2,b3);
   }
```

4. Considere uma fila dupla em alocação sequencial armazenada em um vetor `X` de tamanho M. Fixando-se alguma posição no meio desta fila, ela se divide em duas partes (esquerda e direita) que se comportam cada uma como uma pilha. Assim a fila dupla pode ser simulada por duas pilhas.

   Escreva os algoritmos para a inserção e remoção de elementos da fila dupla considerada desta maneira. Quando uma das pilhas ficar vazia enquanto a outra ainda

não está vazia e se tenta remover um elemento da pilha vazia você deve movimentar itens (ou seja mudar o lugar onde a fila foi quebrada) antes da remoção.

Obs: Suponha existentes as funções *UNDERFLOW* e *OVERFLOW*. Lembre-se que a função *OVERFLOW* só deve ser chamada quando não houver mais espaço disponível no vetor.

5. Faça as seguintes funções para a seguinte estrutura de lista ligada:

```
typedef struct no1{
           int info;
           struct no1 *prox;
        } node, *apontnode;
```

   (a) Uma função que dada uma sequência terminada por zero devolve uma lista ligada com os elementos lidos;

   (b) Faça uma função que insere um elemento no início da lista;

   (c) Faça uma função que dados p remove o elemento na posição p (a partir do primeiro elemento) da lista;

   (d) Faça uma função onde dada uma lista retorna duas listas, uma com os números pares e outra com os ímpares;

   (e) Faça uma função que dada uma lista ordenada, e um novo elemento x, insere x na posição certa da lista.