

MC202 - Turmas EFGHI

Lista de Exercícios 2

1. Uma fila simétrica é uma estrutura linear, similar a uma fila mas na qual inserções e deleções podem ser realizadas em ambas as extremidades. Assim, o conjunto usual de procedimentos para manipulação de filas simétricas poderia incluir: `InsererNaFrente` `RetiraDaFrente` `InsererNoFinal` `RetiraDoFinal`. Quais as estruturas abaixo permitem implementar estes procedimentos com um número constante de operações para cada chamada:

- vetor circular
- lista simplesmente ligada
- lista ligada circular (com cabeça e sem cabeça)
- lista duplamente ligada

Escreva esses procedimentos para as estruturas que satisfazem essa condição.

2. Faça os exercícios da seção 4.5 da apostila.

Para os próximos 5 exercícios usaremos a seguinte declaração de tipo:

```
Type ApontaNo = ^No;
  No = record
    info : integer;
    esq,dir : ApontaNo;
  end;
```

3. Escreva uma *function* `ContaFolhas(p : ApontaNo) : integer` que conta o número de folhas de uma árvore binária apontada por p .
4. Escreva uma *function* `UltimoPreOrdem(p:ApontaNo):ApontaNo` que devolve o apontador ao último nó da árvore binária não vazia apontada por p , quando percorrida em pré-ordem. Não use recursão nem nenhuma estrutura de dados adicional.
5. Faça um procedimento, *Procedure* `Nivel(n:integer)` para visitar todos os nós de um dado nível n de uma árvore binária apontada por p .
6. Escreva uma *function* `Igual(p,q : ApontaNo) : boolean` que verifica se duas árvores binárias apontadas por p e q são iguais, isto é têm a mesma estrutura e os mesmos conteúdos no campo de informação.
7. Escreva uma *function* `Altura(p : ApontaNo) : integer` que calcula a altura (profundidade) de uma árvore binária apontada por p .

8. Faça os exercícios da seção 6.5 da apostila.
9. Suponha que os nós da árvore binária acima são ampliados com um campo marca:

```
Type ApontaNo = ^No;
  No = record
    marca : boolean;
    info  : integer;
    esq,dir : ApontaNo;
  end;
```

Escreva a *function* $Arvore(p : ApontaNo) : boolean$ que verifica se a estrutura de dados apontada por p constitui uma árvore binária.

10. Considere uma árvores binária plena com n nós, representada seqüencialmente por nível num vetor (como visto em sala). Escreva a *function* $SucPreOrdem(i:integer):integer$ que devolve o índice para o nó sucessor, em pré-ordem, do nó indicado por i . Caso o sucessor não exista a função deve devolver o valor zero. Não use recursão nem nenhuma estrutura de dados adicional.