

MC202 — Estruturas de Dados

Professor

Ricardo da Silva Torres

<rtorres@ic.unicamp.br>

Quarta Lista de Exercícios

Nas questões abaixo, considere a seguinte estrutura de dados usada para representar árvores binárias.

```
typedef struct No{
    int valor;
    struct No *dir;
    struct No *esq;
} Arvore;
```

1. Escreva uma função em C que determina a altura de uma árvore binária.

```
int altura (Arvore *arv);
```

2. Escreva uma função em C que libera todos os nós de uma árvore binária.

```
void liberaArvore (Arvore **arv);
```

3. Escreva uma função em C que conta o número de nós do tipo folha.

```
int contaFolhas(Arvore *arv);
```

4. Prove que altura de uma árvore binária balanceada é igual a $\log_2(N)$, onde N é o número de nós.

5. Escreva uma função em C que determina se uma árvore binária é de busca.

6. Uma árvore é equilibrada se, para cada nó, o número de nós da sub-árvore esquerda é igual ao número de nós da sub-árvore direita. Escreva uma função em C que determina se uma árvore binária é equilibrada.

7. Considere o uso de árvores binárias para representar expressões matemáticas (veja a estrutura e a figura a seguir). Escreva uma função que avalia uma árvore.

```
typedef enum {tipo_char, tipo_float} tipo_no;
```

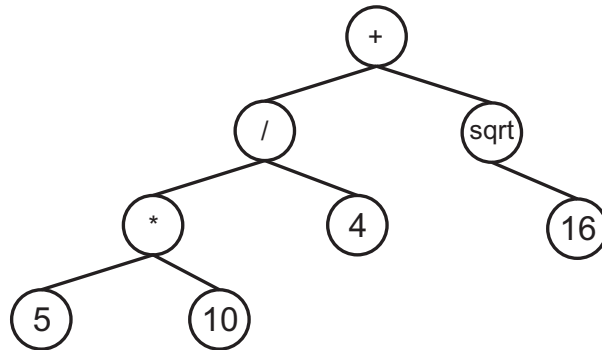
```
typedef struct No {
    union {
        float valor;
        char oper;
    } info;
    tipo_no tipo;
```

```

    struct No *dir;
    struct No *esq;
} ArvoreMat;

float avalia(ArvoreMat *arv);

```



8. Considere a estrutura definida na questão anterior. Escreva uma função que recebe uma expressão matemática em notação pós-fixa e constrói uma árvore binária.

```

ArvoreMat *constroiArvoreMat(char *posfixa);

```

9. Escreva uma função em C que constrói uma árvore binária a partir de percursos *In*-ordem e Pré-ordem armazenados em vetores de tamanho *n*.

```

Arvore *constroiArvore(int *inordem, int *preordem, int n);

```

10. Escreva uma função em C que faça um percurso em profundidade do tipo Pós-ordem sem usar recursão.

```

void posordem(Arvore *arv);

```