

91p

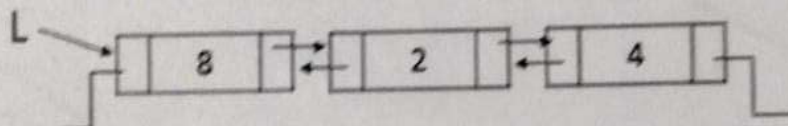
2ª Avaliação Parcial
Curso: Engenharia da Computação
Disciplina: Estruturas de Dados
Prof. Jarbas Joaci de Mesquita Sá Junior
Universidade Federal do Ceará – UFC/Sobral

Nome: Wendel Luan Sousa Batista Data 15/06/2023

1ª. Considere que um nó de uma lista duplamente encadeada é dado por:

```
typedef struct lista_dupl ListaDupl;  
struct lista_dupl { int info; ListaDupl *ant; ListaDupl  
*prox;};
```

Ex:



Implemente uma função que insira um valor em um **novo nó** no fim de uma lista duplamente encadeada. Essa função deverá receber o endereço do começo da lista e o valor a ser inserido, e deverá retornar o endereço do primeiro nó da lista. O protótipo da função deve ser: (2,0 pontos)

```
Lista* insere_fim_lista_dupl(ListaDupl* l, int x);
```

2ª. Considere que um nó de uma árvore binária de busca é dado por:

```
typedef struct arvb ArvB;  
struct arvb { int info; ArvB *esq; ArvB *dir;};
```

Implemente uma função que **conte** quantos nós com o campo **info** com valor **par** existem em uma árvore binária de busca. O protótipo da função deve ser: (2,0 pontos)

```
int qtd_nos_arv(ArvB* a);
```

3ª. Insira os elementos 67, 22, 93, 55, 31, 17, 88, 45, 96, 13, 33, 54 em uma árvore binária de busca. A seguir, remova os elementos 22, 17 e 31. (2,0 pontos)

4ª. Insira os elementos 19, 22, 33, 55, 44, 21, 56, 11, 77, 41 em uma árvore AVL. (2,0 pontos)

5ª. Insira os elementos 11, 62, 33, 12, 51, 78, 11, 81, 93 em uma árvore rubro-negra. (2,0 pontos)

Obs. as inserções e remoções em árvore devem obrigatoriamente ocorrer na ordem apresentada.

AP2 ED 2023.1

Wendel Luon Souza Batista - 516707

1)

```
LISTADUPL * INSERE_FIM_LISTA_DUPL(LISTADUPL * l, INT x) {
```

```
    LISTADUPL * LN = (LISTADUPL *) MALLOC(sizeof(LISTADUPL));
```

```
    LN->INFO = x;
```

```
    LN->PROX = NULL;
```

```
    LN->ANT = NULL;
```

```
    IF (l == NULL)
```

```
        RETURN LN;
```

```
    LISTADUPL * lAux = l;
```

```
    WHILE (lAux->PROX != NULL) {
```

```
        lAux = lAux->PROX;
```

```
    }
```

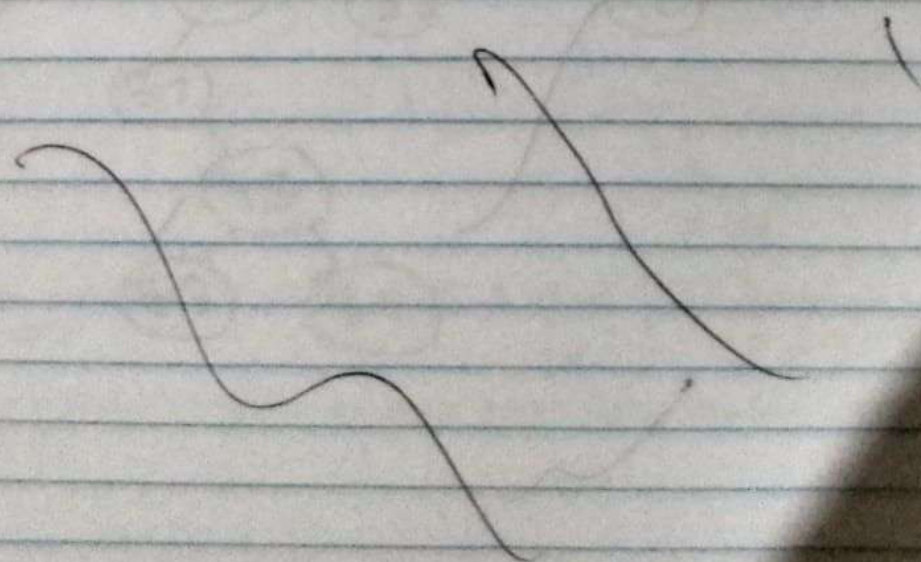
```
    lAux->PROX = LN;
```

```
    LN->ANT = lAux;
```

```
    RETURN l;
```

```
}
```

✓
21/2



02.

```
INT qgd_nod_uvu(A* u, B* a) {
```

```
    IF (u == NULL) {
```

```
        RETURN 0;
```

```
    } ELSE IF (u->info % 2 == 0) {
```

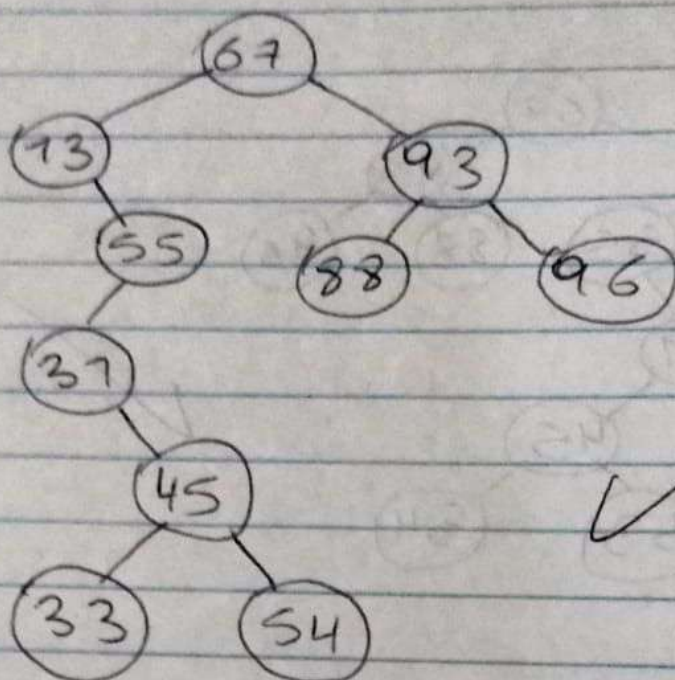
```
        RETURN 1 + qgd_nod_uvu(a->eng) + qgd_nod_uvu(a->din);
```

```
    }
```

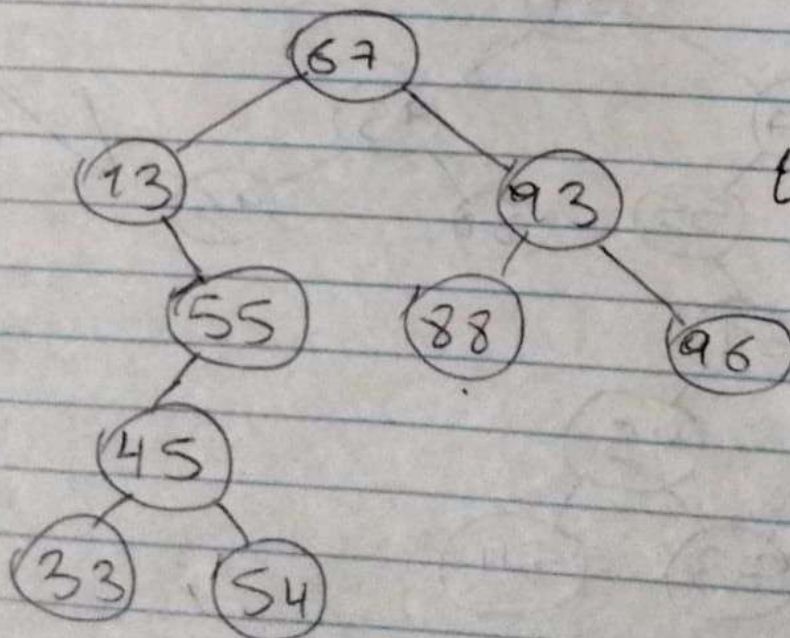
```
    RETURN qgd_nod_uvu(a->eng) + qgd_nod_uvu(a->din);
```


03

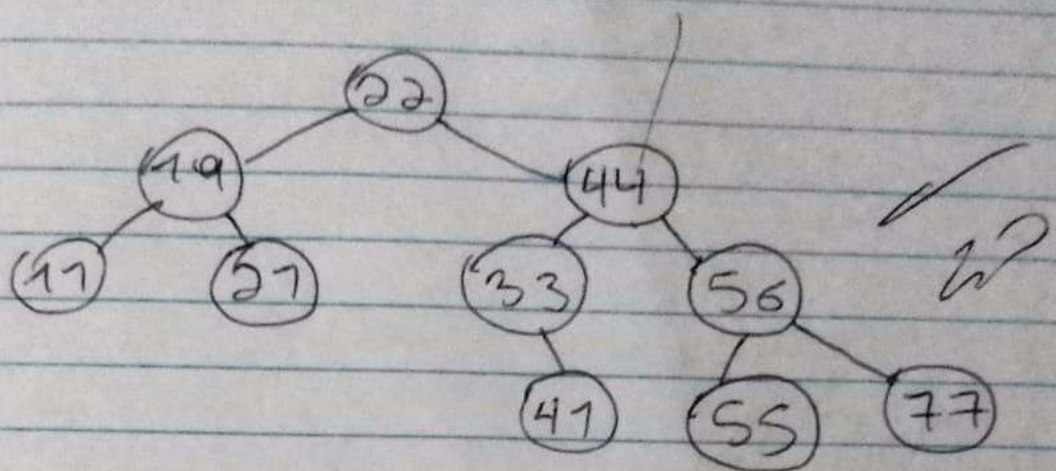
REMOÇÃO do 17:



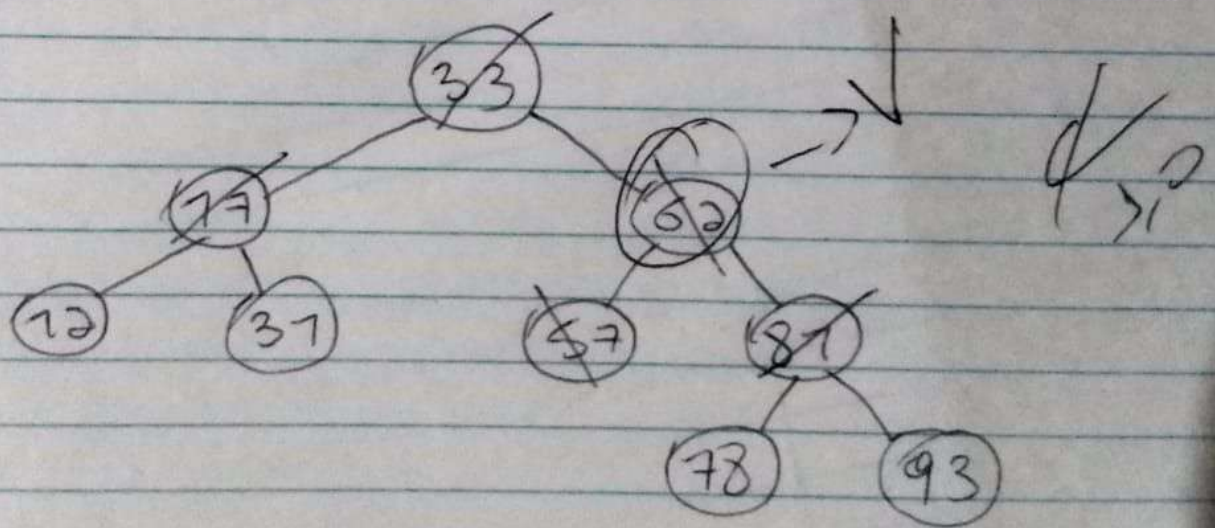
REMOÇÃO do 31:



04. ETAPAS NO RASCUNHO



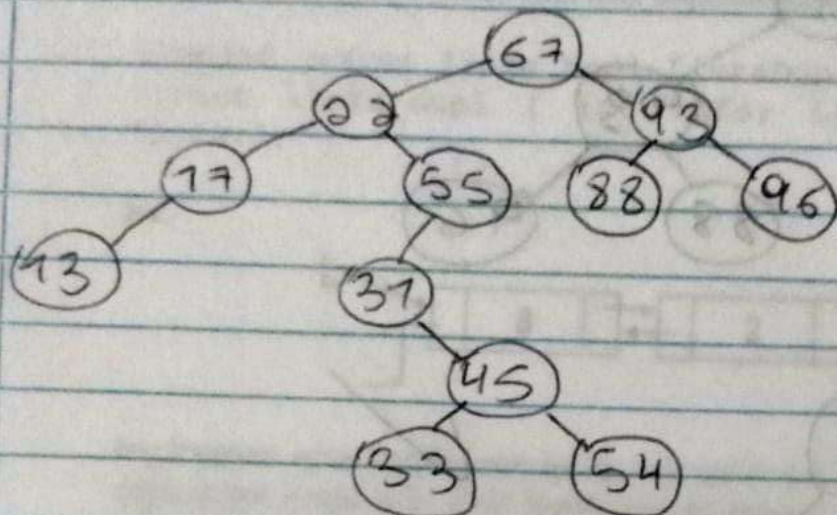
05. ETAPAS NO RASCUNHO



Wendel Luis Sousa Ribeiro.

03.

INSERÇÃO



REMOÇÃO do 22:

