Universidade Federal do Rio de Janeiro
Instituto de Computação
Estruturas de Dados

07/12/2022

1.1– Insira em uma arvore rubro-negra, inicialmente vazia, as seguintes chaves, nesta ordem:
30, 42, 10, 35, 55, 50, 45. Mostre a estrutura antes e depois de cada rotação que for necessária.

1.2- Repita as operações acima em uma arvore **rubro-negra left-leaning**, também inicialmente vazia.

2.1- Considere uma heap binomial,**sem** avaliação tardia, inicialmente vazia. Insira as chaves **7, 15, 12, 2, 6, 3, 4, 11, 5, 7 e 35** nesta ordem.

2.2- Da heap obtida, remova o elemento de prioridade mínima.

3- Construa uma arvore de Huffman para a seguinte frase: "sharks like shirts ". Qual o tamanho desta frase se codificada utilizando sua arvore? (não esqueca do caracter espaco!)

4.1- Insira as seguintes chaves: **6, 41, 33, 17 , 22 e 7** em uma tabela hash com encadeamento interno com área de colisão com espaço para 5 chaves. Utilize a função de hash h(x) = x mod 5.

4.2- Escreva o algoritmo de remocao de uma chave x nesta tabela hash. Lembre-se de atualizar a Lista de Espaco Disponivel (LED).

5- Considere 10 conjuntos inicialmente unitários, com elementos 1, 2, 3, ... 10. Realize as seguintes operações, com compressão de caminhos e união por rank. Mostre o rank de cada elemento.

Une(1,2), une(1,3), une(4,5), une(6,7), une(8,9), une(6,9), une(2,6), une(3,4).
Lembre-se que as operações de união incluem operações find (com compressão de caminhos !).

INSTITUTO DE COMPUTAÇÃO
Universidade Federal do Rio de Janeiro
Centro de Ciências Matemáticas e da Natureza

Nº

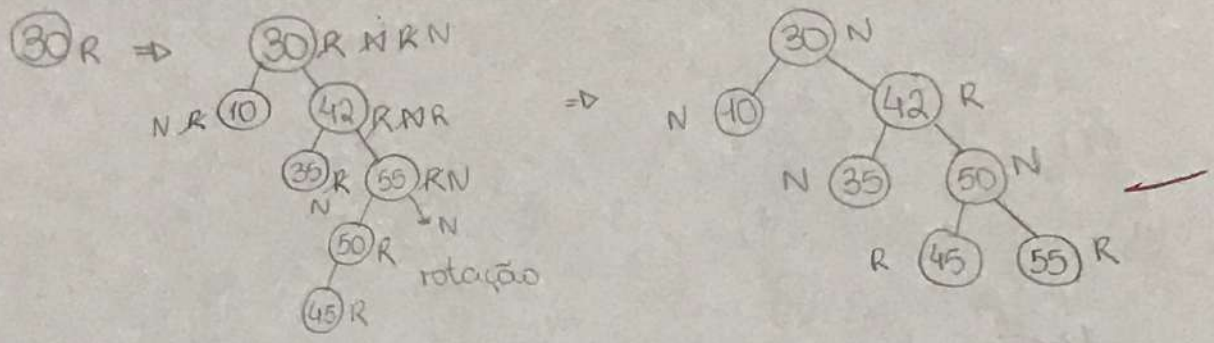
1	2,0
2	2,0
3	2,0
4	1,7
5	2,0
	9,7

Aluno: Caroline Bernat Rebelo **Data:** 07/12/22

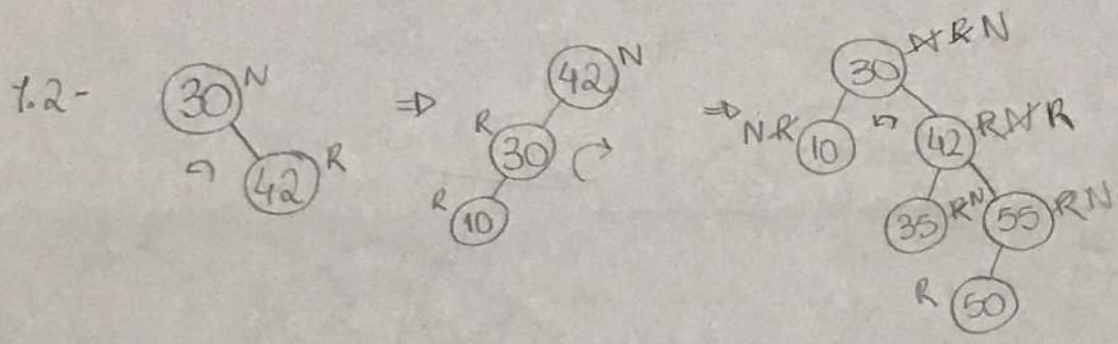
Assinatura: Caroline Bernat Rebelo **Matrícula:** 118046249

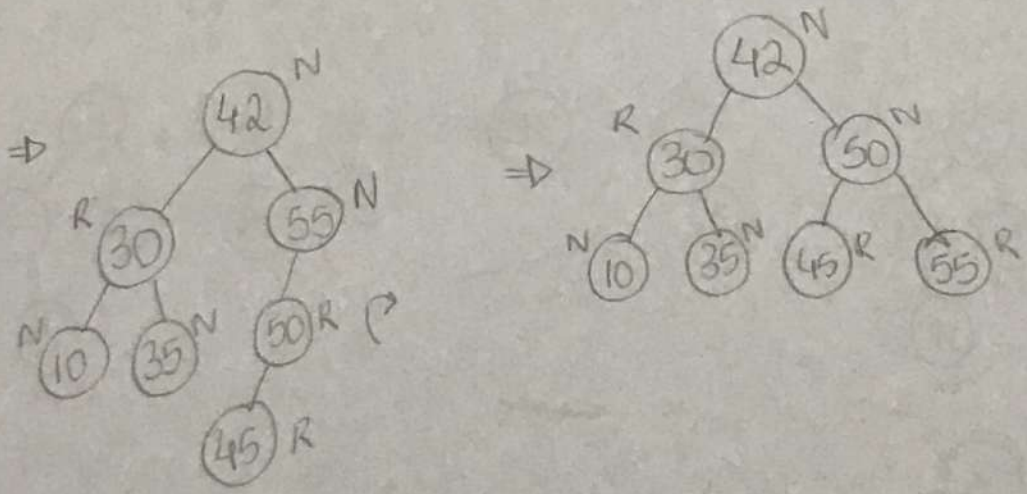
Disciplina: Estruturas de Dados **Turma:**

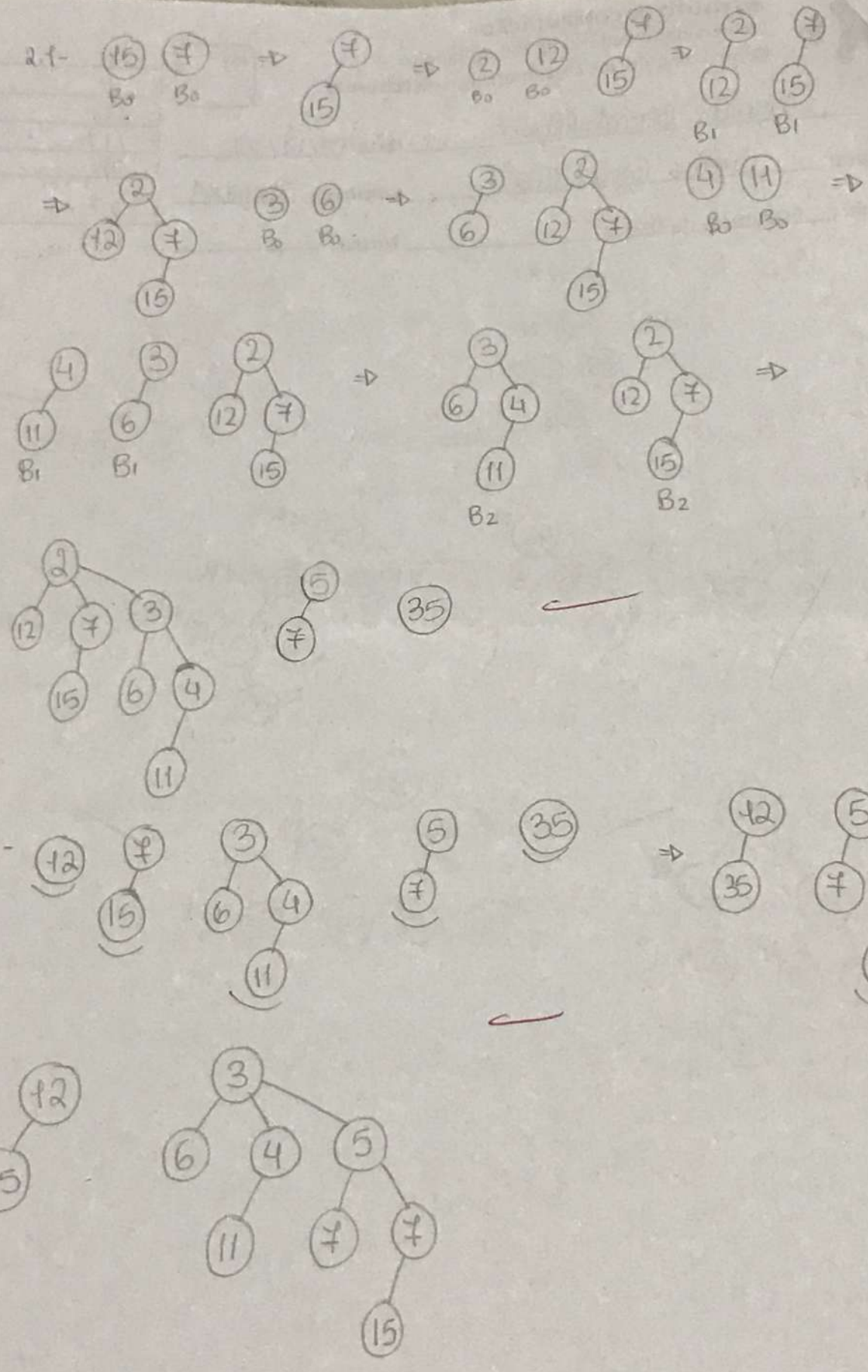
1.1 -



1.2 -





2.1- 
15 7
B0 B0
7
15
2 12
B0 B0
4
15
2 7
12 15
B1 B1
2
12 7
15
3 6
B0 B0
3 2
6 12 7
15
4 11
B0 B0
4 3 2
11 6 12 7
B1 B1 15
3
6 4
11
B2
2
12 7
15
B2
2
12 7 3
15 6 4
11
5
7
35

2.2-
12 7
15
3
6 4
11
5
7
35
12 5
35 7 7
15
12
35
3
6 4 5
11 7 7
15

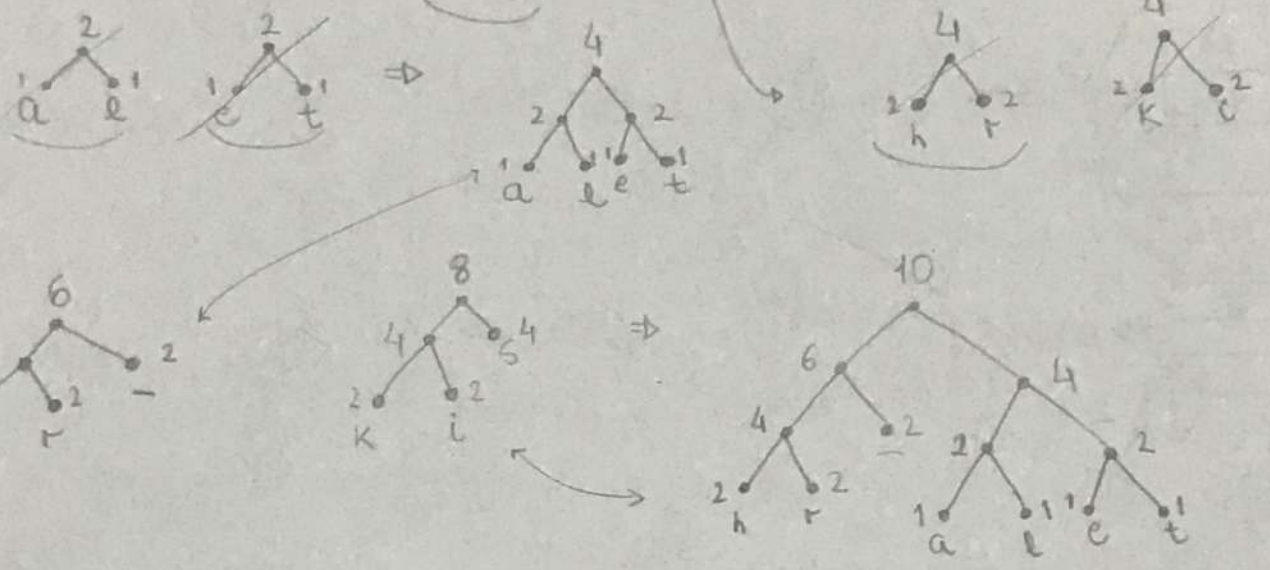
3- ~~a = 1~~ ~~h = 2~~ ~~5 = 4~~

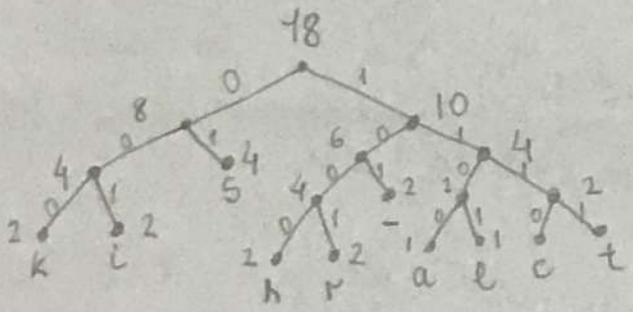
~~l = 1~~ ~~r = 2~~

~~e = 1~~ ~~k = 2~~

~~t = 1~~ ~~i = 2~~

~~_ = 2~~





a = 1100

l = 1101

e = 1110

t = 1111

h = 1000

r = 1001

K = 000

i = 001

_ = 101

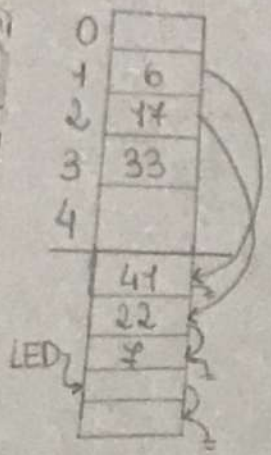
5 = 01

4 + 4 + 4 + 4 + 2.4 + 2.4 + 2.3 +

2.3 + 2.3 + 4.2 = 58

4.1 -

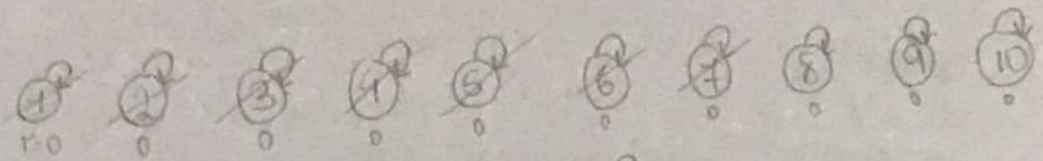
h(6) = 1, h(41) = 1, h(33) = 3, h(47) = 2, h(22) = 2, h(7) = 2



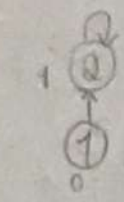
4.2 -

```
remover(tabela, x, tam)
  pos = h(x, tam)
  pt = tabela[pos]
  enquanto pt != NULL e pt → chave != x
    ant = pt
    pt = pt → prox
  se pt == NULL
     Erro, não está na tabela
  se pt → chave == x
    ant → prox = pt → prox
    pt → prox = led
    led = pt
    pt = NULL
```

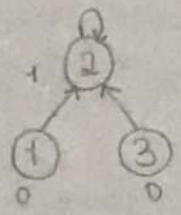
5-



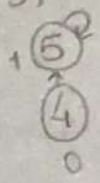
une(1,2):



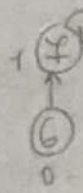
une(1,3):



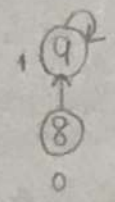
une(4,5):



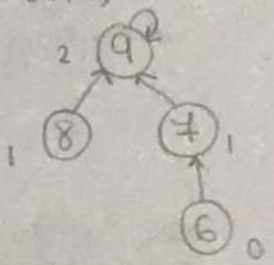
une(6,7):



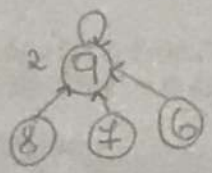
une(8,9):



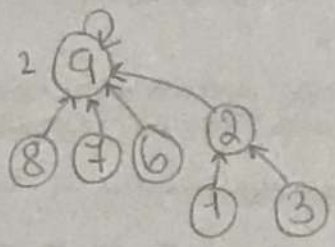
une(6,9):



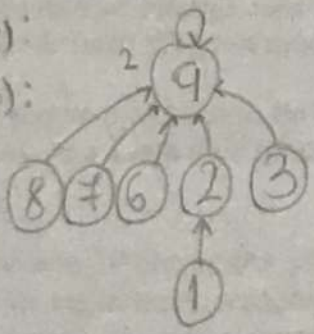
une(2,6):
busca(6):



=>



une(3,4):
busca(3):



=>