Observações:

1. Prova individual e sem consulta.
2. A prova vale 10 pts.
3. Crie um txt ou doc para enviar as respostas teóricas.
4. Mande os códigos .c das demais questões.

**1) ** **(1,5 pt)** De acordo com os valores abaixo, monte a árvore binária de busca. Obs: Não precisa se preocupar com balanceamento da árvore.

# 35, 20, 51, 18, 25, 45, 60,55

# 35

# 20 51

# 18 25 45 60

# 55

**2) ** **(2,5 pt)** De acordo com o trecho de código do arquivo q2\_arvoreBuscaBinaria.c, complete o código para inserir um elemento em uma árvore binária de busca.

**3) ** **(1,0 pt)** De acordo com as árvores abaixo indique se existe algum nó desbalanceado. Se sim, indique qual a rotação deve ser aplicada e como a árvore ficará após a rotação.

* 1. (0,5 pts)



2

10

8

R: sim, está desbalanceado. Deve ser feita uma rotação dupla para a esquerda(RDE).

8

2 10

* 1. (0,5 pts)



8

5

1

R: sim, está desbalanceada. Deve ser feita uma rotação simples para a direita(RSD).

5

1 8

**4) ** **(2,0 pt)** Monte a tabela hash de acordo com os valores abaixo. Para tratamento de colisão. Deve-se utilizar encadeamento fechado (ou listas encadeadas nas colisões). Considere o tamanho 5 no método da divisão.

hash(valor) = x mod 5

# 22, 50, 28, 5, 23, 18, 20

22%5=2

50%5=0

28%5=3

5%5=0

23%5=3

18%5=3

20%5=0

0 = [50, 5, 20]

1 =

2 = [22]

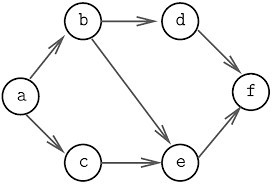
3 = [28, 23, 18]

4 =

**5) ** **(3,0 pt)** Dado o grafo abaixo, apresente:

Obs: Começar do nó a e seguir ordem alfabética.

1. (1,0 pt) Saída utilizando a busca em largura
2. (1,0 pt) Saída utilizando a busca em profundidade
3. (1,0 pt) Matriz adjacente



Saída utilizando a busca em largura :

a,b,c,d,e,f

Saída utilizando a busca em largura :

a,b,d,f,e,c

Matriz adjacente :

A B C D E F

A 0 1 1 0 0 0

B 0 0 0 1 1 0

C 0 0 0 0 1 0

D 0 0 0 0 0 1

E 0 0 0 0 0 1

F 0 0 0 0 0 0

**6) ** **BÔNUS (2,0 pts)** Desenvolva o algoritmo da tabela hash. O algoritmo deve ter uma função para indicar o índice correto, de acordo com o método da divisão. O tamanho do vetor deve ser indicado pelo usuário.

Boa prova !