

# MC202 - Estruturas de Dados

IC - UNICAMP

Prof.: Neucimar J. Leite

Nome: ..... NEUCIMAR .....

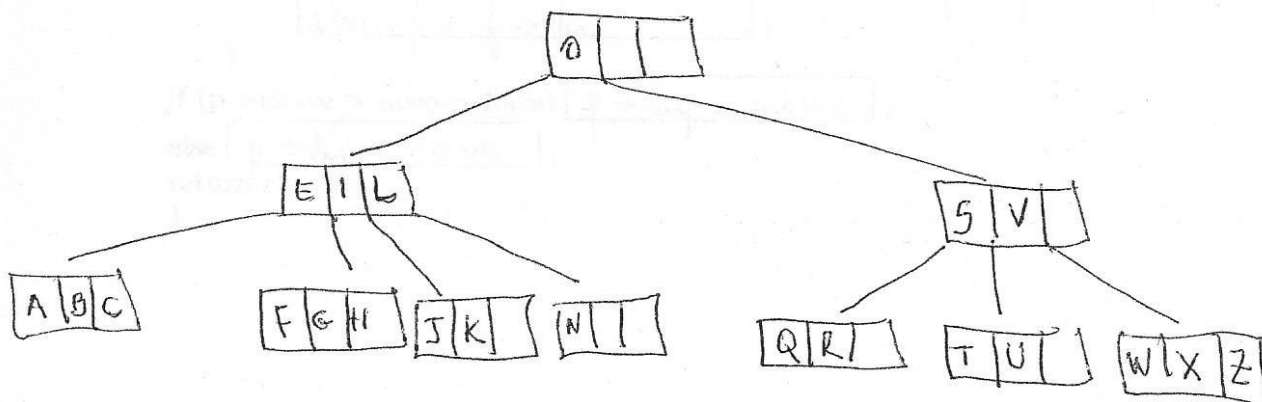
Registro acadêmico: .....

Turma: .....

11 de junho de 2013

1. (2, 0) Desenhe a árvore B de ordem quatro (3 chaves por página) resultante da inserção sequencial das seguintes chaves (atenção: na fase de *splitting* a chave a ser promovida deve ser a mais à esquerda da nova subárvore criada):

NSOAEHBHIFKLQRTVUWZCGJX



2. (2, 0) Complete o programa a seguir que recebe o endereço *novo* de um nó criado e insere iterativamente este novo elemento numa **árvore binária de busca**.

```
typedef struct No {  
    int chave ;  
    struct No *esq, *dir ;  
} ArvBin, NoArvBin ;  
...
```

```
ArvBin *InserI (ArvBin *r, NoArvBin *novo) {  
    NoArvBin *f, *p ;
```

```
    if (r == NULL) return novo ;
```

```
    f = r ;
```

```
    while ( f != NULL ) {
```

```
        p = f ;
```

```
        if ( f->chave > novo->chave ) ; f = f -> esq ;
```

```
        else f = f->dir ;
```

```
    }
```

```
    if (p->chave > novo->chave) p->esq = novo ;
```

```
    else p->dir = novo ;
```

```
    return r ;
```

```
}
```

3. (2, 0) Dado o conjunto de registros abaixo, crie os arquivos de índices necessários para a realização da seguinte consulta: *Alunos: Cidade = Campinas, Curso = Computação*. A inserção dos registros nas listas invertidas criadas deve levar em conta a ordem dos registros na tabela. Estas listas devem ter endereços 0, 1, 2, 3, ...

NRR	RA	Cidade	Curso	CR
1	ra004	Limeira	Civil	0.9
2	ra001	Campinas	Computação	0.8
3	ra003	Campinas	Mecatrônica	0.6
4	ra006	Valinhos	Mecatrônica	0.8
5	ra005	Santos	Computação	0.9
6	ra002	Valinhos	Mecânica	0.7
7	ra007	Campinas	Computação	0.6

Índice Primário

ra001	2
ra002	6
ra003	3
ra004	4
ra005	5
ra006	4
ra007	7

Cidade

Campinas	6
Limeira	0
Santos	4
Valinhos	5

0	ra004	-1
1	ra001	-1
2	ra003	1
3	ra006	-1
4	ra005	-1
5	ra002	3
6	ra007	2

Curso

Civil	0
Computação	6
Mecânica	5
Mecatrônica	3

0	ra004	-1
1	ra001	-1
2	ra003	-1
3	ra006	2
4	ra005	1
5	ra002	-1
6	ra007	4

$$\begin{aligned}
 & \{ra001, ra003, ra007\} \cap \\
 & \{ra001, ra005, ra007\} = \\
 & = \{ra001, ra007\}
 \end{aligned}$$

4. (2,0) Dada uma função de hashing que para cada uma das chaves abaixo atribui os seguintes endereços:

Chave	Endereço atribuído
Anido	20
Cláudia	21
Heloísa	20
Rogério	21
Tomasz	24
Arnaldo	20

Mostre o arquivo final, após a inserção destas chaves, e calcule o número médio de busca. Para tanto, considere os seguintes métodos de inserção de registros: a) (1,0) Overflow progressivo; b) (1,0) Área separada de overflow. Considere, ainda, que o tamanho do arquivo é sempre maior que o número de registros a serem inseridos.

a)

20	Anido (1)
21	Cláudia (1)
22	Heloísa (3)
23	Rogério (3)
24	Tomasz (1)
25	Arnaldo (6)

$$\frac{1 + 1 + 3 + 3 + 1 + 6}{6} = \frac{15}{6} = 2,5$$

b)

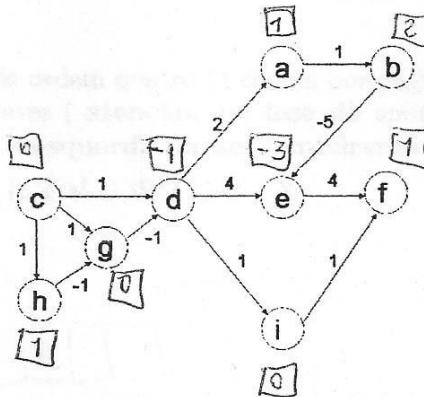
20	Anido → Heloísa → Arnaldo
21	Cláudia → Rogério
24	Tomasz

$$\frac{1 + 2 + 3 + 1 + 2 + 1}{6} = \frac{10}{6} = 1,6$$

5. (2,0) Preencha a tabela abaixo com os valores referentes ao método de correção de rótulos (algoritmo de Lester Ford para o cálculo de caminhos mínimos), visto em sala de aula, para o seguinte grafo com pesos negativos. Para tanto, você deve considerar a seguinte ordem de visita das arestas:

*ab, be, cd, cg, ch, da, de, di, ef, gd, hg, if*

Indique claramente o número de cada iteração e as respectivas correções de distância em cada iteração. Escreva as distâncias finais em cada vértice do grafo.



	início	iteração			
		1	2	3	4
a	$\infty$	3	<del>2</del>	1	
b	$\infty$	$\infty$	4	3	2
c	0				
d	$\infty$	<del>1</del> 0	<del>0</del> -1	-1	
e	$\infty$	<del>0</del> 5	-1	-2	-3
f	$\infty$	<del>2</del> 3	<del>1</del> 2	1	
g	$\infty$	<del>1</del> 0	0		
h	$\infty$	1	1		
i	$\infty$	2	1	0	

