

Universidade Estadual de Santa Cruz  
Ciência da Computação – Estruturas de Dados – 2017.1  
Prática para a prova do 3º Crédito

Nome	Data	Pontos	Nota
		Máx = 100	= Pontos ÷ 10

### Questão 1

1 ponto por chave na posição correta, total 27 pontos

A figura a seguir mostra três vetores, cada um implementando uma **árvore binária de busca** diferente. Note que inicialmente todos os vetores estão vazios, bem como as árvores correspondentes.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
<b>V<sub>1</sub></b>	⊗	94	77	99	54	83	98			66	79	86	97						60			
<b>V<sub>2</sub></b>	⊗	61	40	70	17	50	63	96	10		44		62									
<b>V<sub>3</sub></b>	⊗	56	32	61	20	35			10													

Preencha os vetores acima mostrando o resultado das seguintes operações nas árvores correspondentes.

Árvore	Operações (I = insere, R = remove)
<b>A<sub>1</sub></b>	I(89), I(66), I(83), I(77), I(94), I(54), I(70), I(79), R(66), I(66), I(60), R(89), R(70), I(86), I(99), I(98), I(97)
<b>A<sub>2</sub></b>	I(55), I(33), I(17), I(70), I(96), I(50), I(40), I(10), I(44), I(63), I(62), I(61), R(33), R(55)
<b>A<sub>3</sub></b>	I(47), I(61), I(50), I(26), I(35), I(56), I(29), I(32), I(52), R(47), R(26), R(50), R(29), R(52), I(20), I(10)

### Questão 2

1 ponto por chave na posição correta (1 a 21), total 21 pontos

A figura a seguir mostra três vetores, cada um implementando um **heap** diferente. Note que inicialmente todos os vetores estão vazios, bem como os heaps correspondentes.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
<b>V<sub>1</sub></b>	⊗	1	4	31	14	11	44	36	49	21	54	61	89	71	74	41	94	66	79	26	84	
<b>V<sub>2</sub></b>	⊗	1	4	11	14	21	26	31	36	41	44	49	54	61	66	71	74	79	84	89	94	

Preencha os vetores acima mostrando o resultado das seguintes operações nos heaps correspondente:

Heap	Operações (I = insere)
<b>H<sub>1</sub></b>	I(94), I(89), I(84), I(79), I(74), I(71), I(66), I(61), I(54), I(49), I(44), I(41), I(36), I(31), I(26), I(21), I(14), I(11), I(04), I(01)
<b>H<sub>2</sub></b>	I(01), I(04), I(11), I(14), I(21), I(26), I(31), I(36), I(41), I(44), I(49), I(54), I(61), I(66), I(71), I(74), I(79), I(84), I(89), I(94)

### Questão 3

1 ponto por posição correta nas tabelas, total 33 pontos

Uma técnica de gerenciamento de colisões em tabelas hash com endereçamento aberto é o hashing duplo. Essa técnica usa uma função de hash secundário  $d(k)$  e resolve colisões colocando o item que colidiu na primeira célula disponível na série  $(i + j \cdot d(k)) \bmod N$  para  $j = 0, 1, \dots, N - 1$  onde  $N$  é o tamanho da tabela,  $k$  a chave do item que colidiu, e  $i$  a posição da tabela onde tentou-se inserir o item que colidiu.

Considere uma tabela hash com 11 posições que usa as seguintes funções de hash:

Hash primário:  $h(k) = (2k + 5) \bmod 11$

Hash secundário:  $d(k) = 7 - (k \bmod 7)$

Mostre a tabela hash após a inserção das chaves indicadas abaixo, na ordem em que aparecem. Utilize a tabela na página seguinte para facilitar os cálculos.

- 1) 50, 61, 72, 83, 94, 39, 28, 17, 6, 5, 4
- 2) 2, 13, 23, 34, 44, 55, 65, 76, 86, 97, 98
- 3) 4, 7, 9, 10, 12, 14, 16, 18, 20, 22, 98

0	1	2	3	4	5	6	7	8	9	10
72	6	28	17	5	4	50	83	61	39	94
98	86	97	65	76	44	55	23	34	2	13
14	9	4	10	16	20	18	12	7	22	98

$$k \quad (2k + 5) \bmod 11$$

$$7 - (k \bmod 7)$$

0	5 7	1	7 6	2	9 5	3	0 4	4	2 3	5	4 2	6	6 1	7	8 7	8	10 6	9	1 5
10	3 4	11	5 3	12	7 2	13	9 1	14	0 7	15	2 6	16	4 5	17	6 4	18	8 3	19	10 2
20	1 1	21	3 7	22	5 6	23	7 5	24	9 4	25	0 3	26	2 2	27	4 1	28	6 7	29	8 6
30	10 5	31	1 4	32	3 3	33	5 2	34	7 1	35	9 7	36	0 6	37	2 5	38	4 4	39	6 3
40	8 2	41	10 1	42	1 7	43	3 6	44	5 5	45	7 4	46	9 3	47	0 2	48	2 1	49	4 7
50	6 6	51	8 5	52	10 4	53	1 3	54	3 2	55	5 1	56	7 7	57	9 6	58	0 5	59	2 4
60	4 3	61	6 2	62	8 1	63	10 7	64	1 6	65	3 5	66	5 4	67	7 3	68	9 2	69	0 1
70	2 7	71	4 6	72	6 5	73	8 4	74	10 3	75	1 2	76	3 1	77	5 7	78	7 6	79	9 5
80	0 4	81	2 3	82	4 2	83	6 1	84	8 7	85	10 6	86	1 5	87	3 4	88	5 3	89	7 2
90	9 1	91	0 7	92	2 6	93	4 5	94	6 4	95	8 3	96	10 2	97	1 1	98	3 7	99	5 6

### Rascunho

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22