

AED 3

Segunda lista de exercícios

1. Se atualizarmos um campo que não é chave (ex: endereço, idade, etc.) de um registro de tamanho variável de um arquivo indexado, haverá alguma situação em que o índice também deverá ser atualizado? Justifique.
2. Cite duas vantagens e duas desvantagens dos arquivos indexados sobre os arquivos sequenciais.
3. Crie uma árvore B de ordem 3 vazia e, em seguida,
 - a. Acrescente a ela as seguintes chaves: A, L, G, O, R, I, T, H, M, e S.
 - b. Remova, da árvore resultante, as seguintes chaves: L, G, H, I, R e A.
4. Crie uma árvore B+ de ordem 5 vazia e, em seguida,
 - a. Acrescente a ela as seguintes chaves: 9, 5, 1, 7, 11, 13, 8, 6, 12, 3, 10, 2, 0, 15, 4 e 14.
 - b. Remova, da árvore resultante, as seguintes chaves: 7, 9, 3, 2 e 5. Priorize as fusões com e as cessões de chaves de irmãos esquerdos.
5. Se uma página que não é folha de uma árvore B possui 7 chaves, quantos filhos ela possui? Justifique a sua resposta.
6. Em uma árvore B de ordem 30, qual é o número mínimo e o número máximo de elementos em cada página da árvore? Esse valor é válido para todas as páginas sem exceção? Caso negativo, justifique.
7. Considere uma tabela *hash* com 17 endereços. Considere a função de dispersão $h(k) = k \bmod 17$. Acrescente as chaves 20, 25, 13, 9, 14, 22, 39, 19, 6, 7 e 33, nesta ordem, considerando o tratamento de colisões por *double hashing*, considerando como segunda função de dispersão $h_2(k) = (k^2 \bmod 16) + 1$.
8. Crie uma tabela *hash* dinâmica com *buckets* de tamanho 3 e acrescente a ela as chaves 20, 25, 13, 9, 14, 22, 39, 19, 6, 7, 33.
9. Cite uma estrutura de dados adequada para armazenamento de listas invertidas em disco. Justifique.
10. Usando um dicionário cujos índices (ou posições) são representados com apenas 6 bits e considerando que o conjunto de símbolos é composto apenas 26 caracteres de A a Z, calcule quanto bits são necessários para a seguinte mensagem compactada com LZW:

ABBBAABACDBBBAABCDDDAABACDBBA
11. Crie a árvore de Huffman para a mensagem abaixo e informe quantos bits são necessários para compactar essa mensagem.

ABBBAABACDBBBAABCDDDAABACDBBA

GABARITO

Questão 1

A alteração de campos que não são chave apenas elimina a necessidade de um índice de chave primária. Assim, se esse campo for usado em um índice secundário, então apenas esse índice precisará ser atualizado.

Se, na alteração, o registro mudar de tamanho, é importante considerar se esse registro (ou qualquer outro) precisará ser reposicionado. Isso geralmente acontece quando a alteração resulta em aumento do tamanho do registro. Nesse caso, qualquer índice direto, especialmente o baseado na chave primária, precisará ser alterado.

Questão 2

VANTAGENS:

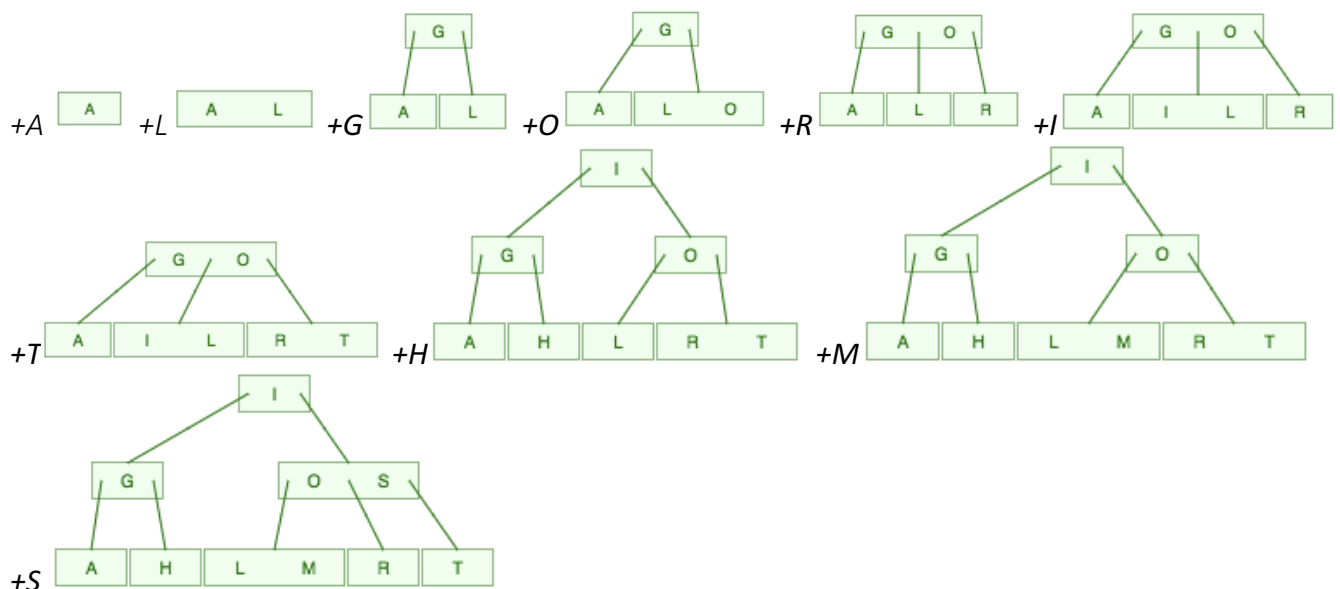
- Os arquivos indexados, por meio de vários índices, permitem a busca rápida por diferentes campos.
- Uma alteração em um registro, não implicará em movimentação de outros registros.

DESVANTAGENS:

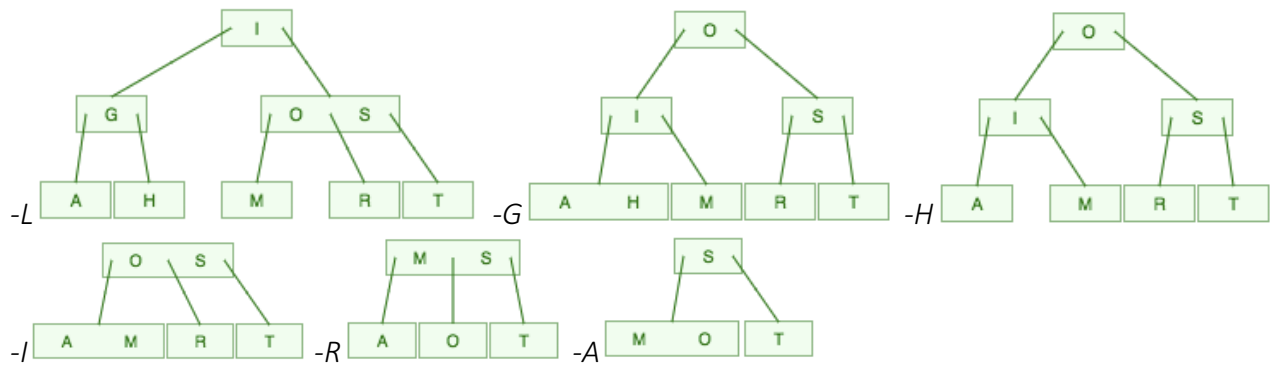
- O processamento sequencial em arquivos indexados é mais lento, pois os registros serão localizados no arquivo de dados individualmente (a partir da ordenação dada pelo índice).
- É necessário manter vários arquivos: um de dados e outro para cada índice.

Questão 3

a)

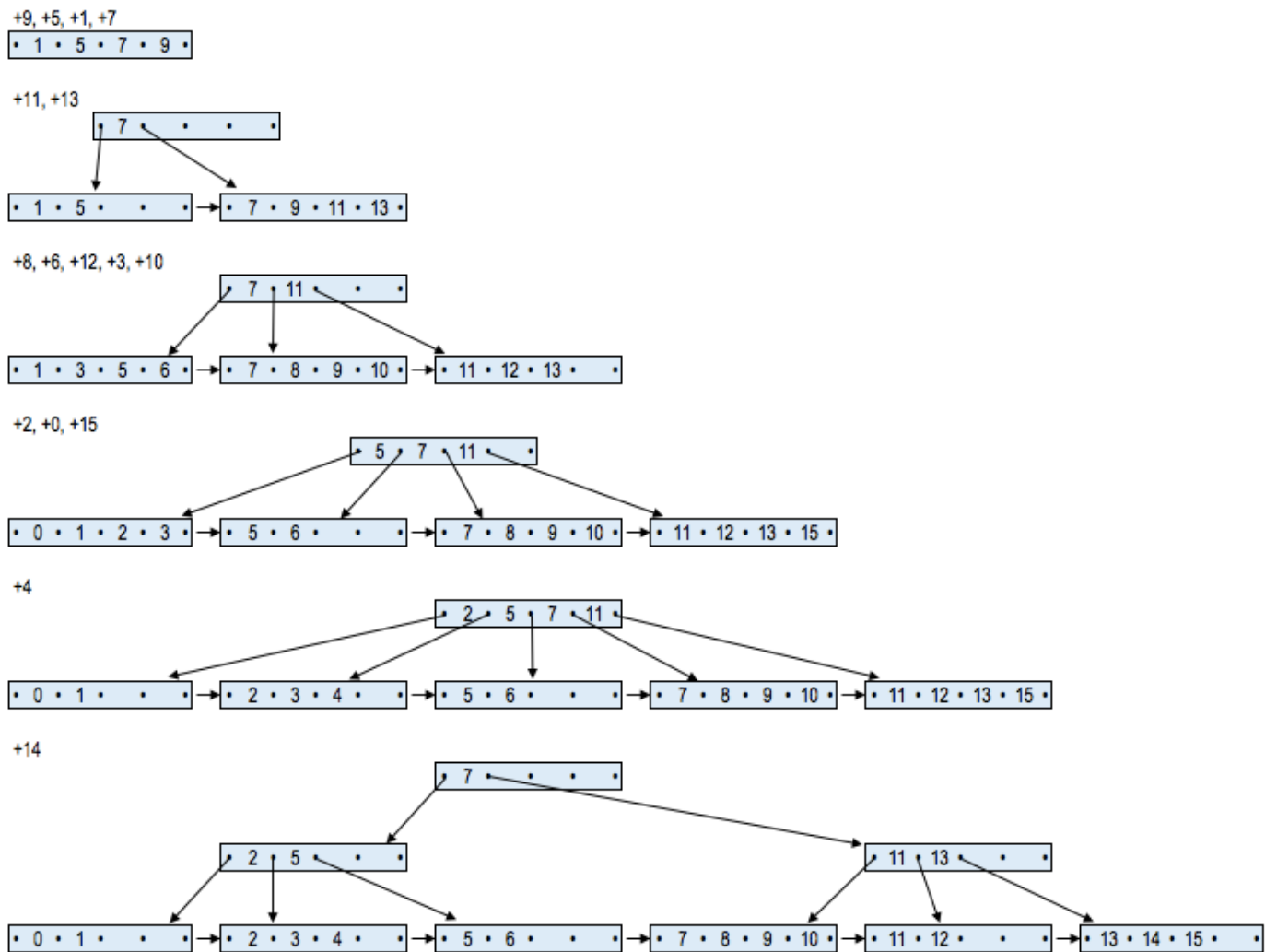


b)



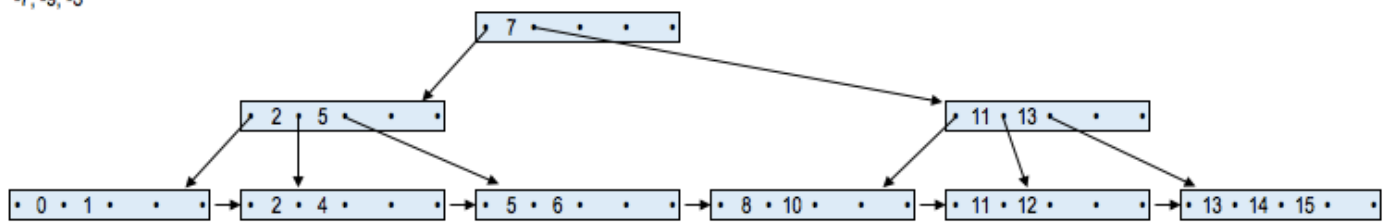
Questão 4

a)

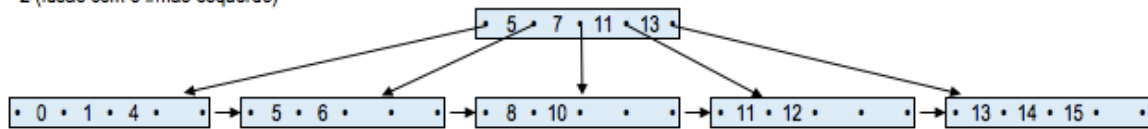


b)

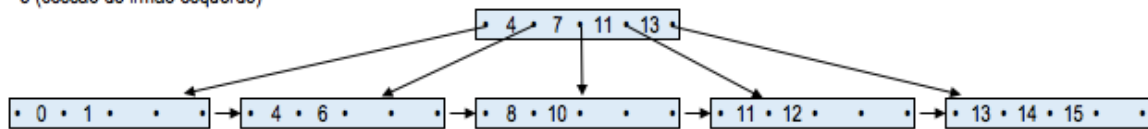
-7, -9, -3



-2 (fusão com o irmão esquerdo)



-5 (cessão do irmão esquerdo)



Questão 5

8. O número de filhos em páginas que não são folhas (exceto a raiz) é sempre o número de chaves + 1.

Questão 6

Em uma árvore B de ordem 30, o número máximo de filhos é 30. Assim, o número máximo de elementos é 29 ($\text{ordem} - 1$) e o número mínimo é 14 (a metade inteira do número máximo). A raiz é a única exceção, em que o número mínimo de chaves é 1.

Questão 7

k	h(k)	$h_2(k)$	Tentativas de endereços
20	3	1	3
25	8	2	8
13	13	10	13
9	9	2	9
14	14	5	14
22	5	5	5
39	5	2	5 - 7
19	2	10	4 - 19
6	6	5	6
7	7	2	7 - 9 - 11
33	16	2	33

	Tab. Hash
0	
1	
2	19
3	20
4	
5	22
6	6
7	39
8	25
9	9
10	
11	7
12	
13	13
14	14
15	
16	33

Questão 8

ESTADO INICIAL

Diretório		Buckets					
p=1		p'	n	0	1	2	
0	0	0	1	0			
1	1	1	1	0			

APÓS INCLUSÃO DE 20, 25, 13, 9, 14 E 22

Diretório		Buckets					
p=1		p'	n	0	1	2	
0	0	0	1	3	14	20	22
1	1	1	1	3	9	13	25

APÓS INCLUSÃO DE 39 E 19

Diretório		Buckets					
p=2		p'	n	0	1	2	
0	0	0	1	3	14	20	22
1	1	1	2	3	9	13	25
2	0	2	2	2	19	39	
3	2						

APÓS INCLUSÃO DE 6 E 7

Diretório		Buckets					
p=2		p'	n	0	1	2	
0	0	0	2	1	20		
1	1	1	2	3	9	13	25
2	3	2	2	3	7	19	39
3	2	3	2	3	6	14	22

APÓS INCLUSÃO DE 33

Diretório		Buckets					
p=3		p'	n	0	1	2	
0	0	0	2	1	20		
1	1	1	3	3	9	25	33
2	3	2	2	3	7	19	39
3	2	3	2	3	6	14	22
4	0	4	3	1	13		
5	4						
6	3						
7	2						

Questão 9

Uma possível estrutura para armazenamento das listas invertidas é uma combinação de uma tabela hash como dicionário e um arquivo contendo múltiplas listas invertidas, como mostra a figura abaixo:

ARQUIVO DE DADOS

Registros

1	Java Web Services
2	Web Design Responsivo
3	Web Services em PHP
4	Programação Java para a Web
5	Desenvolvimento Web Java

TABELA HASH

termos	endereço
a	13
desenvolvimento	15
design	4
em	8
java	0
para	12
php	9
programação	10
responsivo	5
services	2
web	1

LISTAS

End.	Registro	Próximo
0	1	11
1	1	3
2	1	7
3	2	6
4	2	-1
5	2	-1
6	3	14
7	3	-1
8	3	-1
9	3	-1
10	4	-1
11	4	17
12	4	-1
13	4	-1
14	4	16
15	5	-1
16	5	-1
17	5	-1

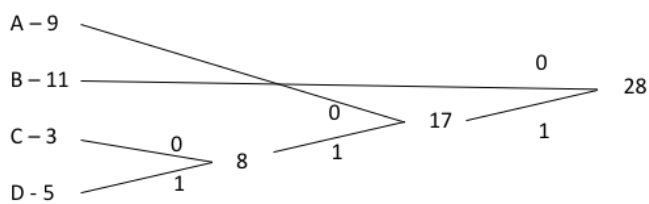
Questão 10

TEXTO:	A	B	B	B	A	A	B	A	C	D	B	B	B	A	A	B	C	D	D	D	A	A	B	C	D	B	B	A
SAÍDA:	0	1	27		0	26		0	2	3	27		1	29		1	32		3	3	36			32		28		

0	A	9	J	18	S	27	BB	36	AAB	45		54		63	
1	B	10	K	19	T	28	BBA	37	BC	46		55			
2	C	11	L	20	U	29	AA	38	CDD	47		56			
3	D	12	M	21	V	30	ABA	39	DD	48		57			
4	E	13	N	22	W	31	AC	40	DA	49		58			
5	F	14	O	23	X	32	CD	41	AABC	50		59			
6	G	15	P	24	Y	33	DB	42	CDB	51		60			
7	H	16	Q	25	Z	34	BBB	43		52		61			
8	I	17	R	26	AB	35	BA	44		53		62			

Serão 18 índices na saída, de 6 bits cada, totalizando 108 bits de mensagem.

Questão 11


$$A - 10 - 2 \text{ bits} * 9 = 18 \text{ bits}$$

B - 0 - 1 bit * 11 = 11 bits

C – 110 – 3 bits * 3 = 9 bits

D – 111 – 3 bits * 5 = 15 bits

Serão necessários 53 bits para compactar a mensagem, sem considerar o armazenamento da árvore ou da tabela.