

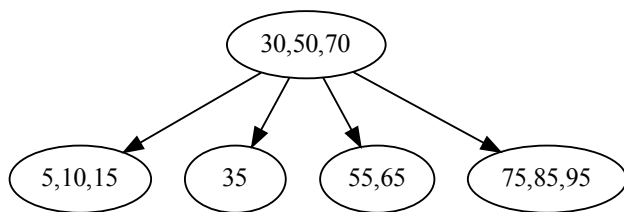
ORGANIZAÇÃO DE ESTRUTURAS DE ARQUIVOS
PROVA 2 - 2022.1 - BCC - PROF. RENATO MAURO

Nome: _____

Questão 1 Em uma árvore B de grau 4, ao inserir 10 chaves, qual é o número máximo de splits que podem ocorrer?

- ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

Questão 2 Considere a árvore B da figura abaixo, de grau 4. Faça o que se pede nos itens de forma independente, ou seja, use sempre a árvore original. Redesenhe em cada um dos itens.



(a) Inserção da chave 12.

(b) Remoção da chave 55.

Questão 3 Qual é o código de Huffman ótimo para os símbolos A:1, B:1, C:2, D:3, E:5, F:8, G:13, H:21? Construa a árvore e especifique o código de cada símbolo.

Símbolo	Código
A	
B	
C	
D	
E	
F	
G	
H	

Para pensar depois de prova: as frequências propostas correspondem à sequência de Fibonacci. Você conseguiria generalizar a sua resposta para uma sequência de tamanho N?

Questão 4 No algoritmo de hash estudado em aula, as colisões eram tratadas por encadeamento externo, ou seja, para cada chave com colisão, um elemento era inserido depois da tabela original, com o devido encadeamento.

- (a) Para a sequência a seguir, crie uma tabela de hash, usando o encadeamento interno. Funcionará da seguinte forma: para cada chave, insira a chave apenas se não houver colisão. Caso haja, reserve a chave para uma segunda rodada. Na segunda rodada, as cha-

ves com colisão serão inseridas, dentro da própria tabela, ao invés de ser fora dela. Utilize como função o resto da divisão por 11. Chaves: 55, 78, 67, 40, 36, 14, 13, 22, 35, 47, 8

POS	CHAVE	PROX
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

(b) Quais vantagens e desvantagens do encadeamento externo e interno?

(c) No exemplo proposto, qual é o número de comparações no pior caso?

Questão 5 Considere que as funções abaixo já se encontrem implementadas na linguagem de programação de sua preferência. Estas funções são utilizadas para manipular uma árvore B+.

- Dado o nome de uma arquivo retorna um objeto *ArvoreB*.

AbreBMais(<nome arquivo>) -> Arvore

- Dado uma árvore aberta e uma chave, retorna a folha onde a chave se encontra.

BuscaPag(<Arvore>,<chave>) -> Pag

- Recupera todas as chaves (inteiros) de uma página em um array de inteiros.

Chaves(<Pag>) -> int[]

- Obtém o tamanho do array de chaves:

Tam(int[]) -> int

- Recupera a página folha vizinha de uma determinada página, ou nulo se não existir.

Vizinho(<Pagina>) -> Pagina

Uma árvore B+ pode ser usada como uma estrutura de dados temporária para se ordenar um arquivo. Considere o seguinte método: para cada chave de ordenação do arquivo, insira a chave e a posição correspondente ao registro. Após inserir todas as chaves, visite as chaves em ordem crescente e copie o registro do arquivo original em um novo arquivo.

(a) Para o arquivo de CEP estudado em aula, ordene-o usando esse método. Faça um esboço de como seria a implementação do algoritmo aqui proposto.

(b) Na solução proposta, estamos usando 3 arquivos: O arquivo original, o arquivo temporário com a árvore e o novo arquivo ordenado. Seria possível usar o um mesmo arquivo, sobrescrevendo o arquivo original e armazenando nele o ordenado? Explique.
