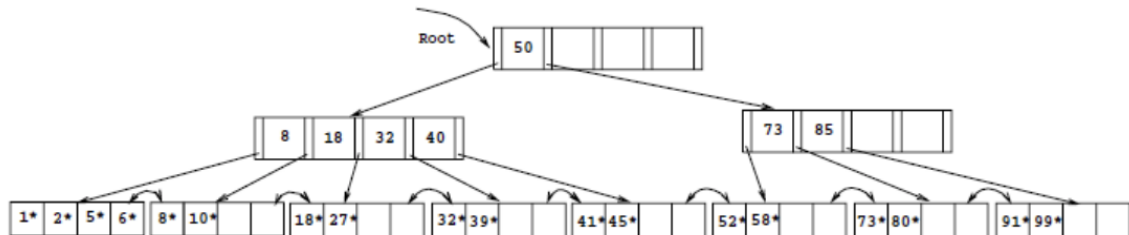


ORGANIZAÇÃO DE ESTRUTURAS DE ARQUIVOS
PROVA 2 - 2022.2 - BCC - PROF. RENATO MAURO

Nome: _____

Questão 1 Considerando a figura abaixo, onde está ilustrada uma árvore B+, faça o que se pede nos itens a seguir. Responda aos itens de forma independente, ou seja, sempre usando a árvore original.



- (a) Insira a chave 9. Indique o número de leituras e escritas necessárias para a realização desta inserção.
- (b) Insira a chave 3. Indique o número de leituras e escritas necessárias para a realização desta inserção.
- (c) Remova a chave 27. Indique o número de leituras e escritas necessárias para a realização desta remoção.
- (d) Remova a chave 80. Indique o número de leituras e escritas necessárias para a realização desta remoção.

Questão 2 Você decidiu usar uma árvore B+ em uma aplicação para indexar determinado arquivo e descobriu que o sistema operacional consegue recuperar blocos de 8192 bytes de forma eficiente. As chaves nessa aplicação ocupam 10 bytes e a referência para blocos (parâmetro do *seek*) ocupam 8 bytes. Suponha que o arquivo a ser indexado tenha em torno de 10^9 registros.

- (a) Qual é o grau mínimo dos nós internos (excluindo as folhas e a raiz)? Justifique.
- (b) Quantos acessos são realizados no máximo nesta árvore para se encontrar uma chave, sabendo que a raiz estará sempre em memória e o primeiro acesso à raiz não é contabilizado?

Questão 3 Considere a estrutura de HASH desenvolvida nas aulas. O arquivo de HASH consiste em uma tabela com 3 colunas: a chave, a posição no arquivo de dados e um ponteiro para o próximo elemento no caso de colisão. Para ilustrar, segue algumas chamadas em Python (fora de ordem e de contexto) do trecho de código utilizado para a manipulação do arquivo de HASH. Use a linguagem de programação de sua preferência.

```
indexName = "data/cep-hash.dat"
fi = open(indexName,"rb")
fi.close()
indexFormat = "8sLL"
indexStruct = struct.Struct(indexFormat)
indexRecord = indexStruct.unpack(fi.read(indexStruct.size))
```

- (a) Escreva um programa que leia toda a tabela sequencialmente e copie para um novo arquivo as chaves e a posição no arquivo de dados de cada uma delas.
- (b) No caso do item anterior, qual a ordem que você supõe que as chaves estarão organizadas?
- (c) Ao analisar o arquivo HASH, observa-se um número excessivo de colisões, pois a tabela está praticamente cheia. Para isso há necessidade de expansão da tabela. Como é feito este procedimento?

Questão 4 Em uma tabela de hash inicialmente vazio, inclua as chaves 72, 47, 16, 44, 83, 21, 20, 37, 35. Obedeça a ordem de inserção. Use como função de HASH resto da divisão por 11. Use o método do encadeamento externo para o tratamento de colisões.

Questão 5 Considere um arquivo contendo chaves e posições ordenado pelas chaves. Proponha uma algoritmo para construção de uma árvore B+ a partir desses dados de forma que a árvore B+ tenha altura mínima. Dica: construa a árvore de baixo para cima, se aproveitando do fato de que o arquivo original já está ordenado pelas chaves. Não precisa ser um programa, basta a explicação detalhada de como o algoritmo deve ser implementado.

BOA PROVA!!