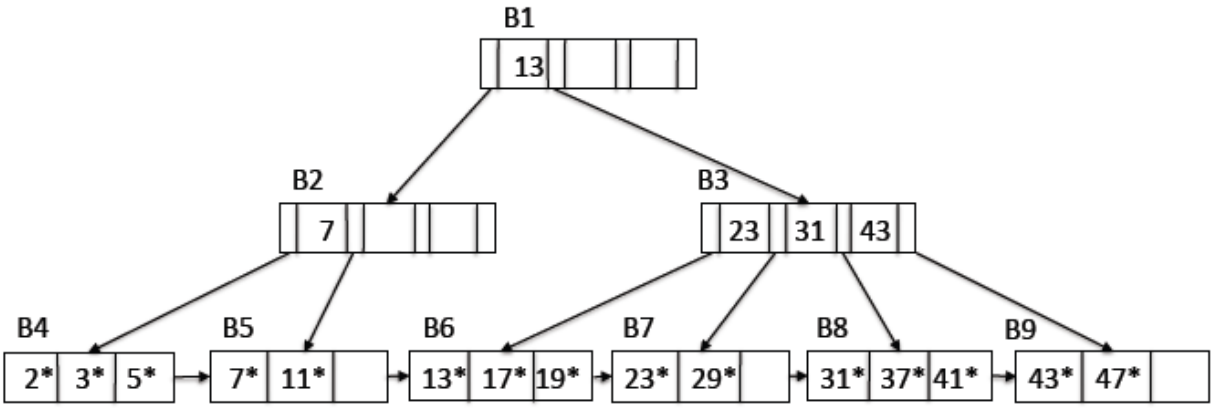
Database Management Systems

Practice Problems: Storage, File Systems, Indexing, Hashing

(1). Consider the following B+ tree, where the index blocks are identified as B1, B2, B3, and the data entry blocks are B4,..., B9. For each of the following lookup operations, describe which blocks would be read and in which order. For each of the insert and delete operations, apply them to the given B+ tree and draw the resulting tree.

(a) Lookup record with key 41.
(b) Lookup record with key 40.
(c) Lookup records with keys in the range 20 to 30.
(d) Insert a record with key 1.
(e) Insert records with keys 14, 15 and 16.
(f) Delete record with key 23.
(g) Delete records with keys 23 and higher



(2). Consider the following relational instance of the Internet Movie Database (IMDb).

imdbID	title	yearReleased	gross(M)	rating
100	'Practical Magic'	1998	46	7
101	'Dark Knight'	2008	532	9
102	'Beetle Juice'	1998	73	7
103	'Heartbreakers'	2001	40	6
104	'Shrek'	2001	267	8
105	'The Simpsons Movie'	2007	183	8
106	'The Holiday'	2006	63	7

(a) Assuming that the hash function *h* takes the gross and rating, and calculates *(gross+rating)* mod 5. For example, let us look at the first tuple in the relation. *h*(46+7) = (53) mod 5 =3. Assume that all hash values fit in one block, and only one record of TITLE can fit in one block. Show a picture of the resulting file organization. (E.g. 3 -> 'Practical Magic')

(b) Assuming that the hash function *h* takes the imdbID and calculates (imdbID) mod 5. For example, let's look at the first tuple in the relation. *h* (100) mod 5 = 0. Assume that all hash values fit in one block, and only one record of TITLE can fit in one block. Show a picture of the resulting file organization.(E.g. 0 -> 'Practical Magic')

(c) Look at the file organization resulting from the above two questions. (1) Which hash function would you prefer to use and (2) why?

(3). For each of the following scenarios, which of the file organizations – heap, B+ tree indexed file, or hash indexed file – would you choose.

(a) The most common operation is searching based on range of field values.

(b) You only want to perform inserts and full file scans.

(c) The most common operation is searching for a record based on a particular field value.

(4). A disk spins at 10,000 rpm, has 256K tracks per surface, an average of 512 sectors per track, and four platters; data is stored on both sides of the platters. The average seek time is 5 ms but the time to seek from one track to an adjacent track is 1 ms. Assume the sector size is 512 bytes and the page size is 16 sectors. Here K=1024.

a) What is the capacity of a platter? What is the capacity of the disk?

b) What is the maximum rotational delay?

c) How many cylinders are required to store a 500MB file?

d) Assume page size is 8 sectors, how many I/Os are required to read the file?

5. Consider the join of two relations r and s, by a nested loop algorithm.

```
for each tuple tr of r do
      for each tuple ts of s do
            if the tuple tr and ts satisfy the join condition
            then output the result
      end for
end for
```

In the above algorithm, we call r an outer relation (in the outer for loop), and s an inner relation (in the inner for loop). Suppose that the relation r is stored in 2 blocks, and the relation s in 3 blocks on disk. All the blocks are in the same size, and in each block there are 3 tuples for both r and s. We consider how to process the algorithm at runtime using the buffer management strategies MRU and LRU.

For every outer relation block, once processing of a block of an outer relation is finished, we can free that block, since we do not need it any more in processing the join operation. Here, by free it means that that block can be replaced by other blocks even when LRU is used. It is called the toss-immediate strategy. The toss-immediate strategy should be used in both LRU and MRU if needed. In MRU, since a block of an outer relation is the most recently used and should not be replaced by MRU, if there are remaining blocks of the inner relation to be processed, it should pin the outer relation block in the buffer, in order to avoid it to be replaced. By pin it means that the block being pinned is not a candidate for replacement until the page is freed. Suppose that there is a buffer which can keep 4 blocks in main memory.

a. Show the process in brief, and give the number of disk accesses using LRU block-replacement scheme.

b. Show the process in brief, and give the number of disk accesses using MRU block-replacement scheme.

c. Give your comments on when you should use LRU and when you should use MRU in brief.