CSC 3170 Assignment 4

This is an individual assignment and should be

submitted by 5 pm, 30 April 2022 via Blackboard

1. Consider a file of N records arranged according to the Zipf distribution, where the probability of choosing the record in position k is inversely proportional to its position:

$$p_k = \frac{C}{k} \qquad\qquad k = 1, 2, \ldots, N$$

i.e. the probability that the record in position k being the required record in a search is given by p_k, and C is a normalizing constant. Thus, the records at the beginning of the file has higher chance of being chosen.

(a) Let X_N be the (random) number of comparisons to locate a given record present in the file of N records. Derive an expression for the average $E(X_N)$ for the Zipf distribution (the expression should not include the constant C).

(b) Compare the search performance of the Zipf distribution against the uniform distribution (i.e. $p_k = 1/N$) for $N = 10$. (Note: since N is not large here, the exact formula should be used).

(c) Determine the file size N^* where $E(X_{N^*})$ is the same for both the Zipf and the uniform distribution.

(d) Consider now the file of N records are arranged in ascending order of the primary key, and assuming the required record is not present in the file. What is the approximate average number of comparisons required to conclude the search?

(e) Consider the file of N records are now arranged using the heap organization where records are not arranged in any particular order. Determine, under the uniform distribution, the average number of comparisons required to conclude the search for (i) when the required record is present in the file, and (ii) when the required record is not present in the file.

2. Consider a B-tree where each node can have a maximum of 23 children. Assume that each node of the B-tree is of average fullness. Consider that the root node is at Level 0. Determine for each of the following levels, the average number of nodes, the average number of key entries, and the average number of children pointers for that level:

(i) Level 1
(ii) Level 3
(iii) Level 4

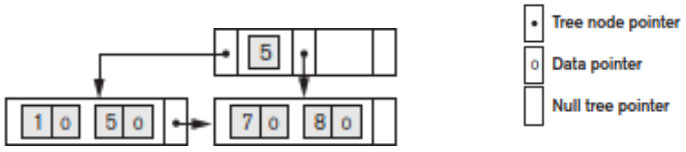
(Note: you may round to the nearest integer in determining the number of children pointers)

Determine the average number of entries that such a tree holds, assuming that:

(iv) the height of the tree is 2 (i.e. the tree consists of Level 0, Level 1, and Level 2)
(v) the height of the tree is 3
(vi) the height of the tree is 4.

From the above observations or otherwise, derive a general formula for the average total number of entries that such a tree holds for a given height *h*.

3. Consider the following B^+-Tree.

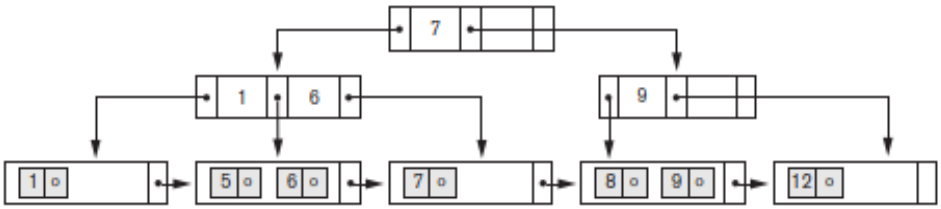


Show each of the tree structure after the insertion of

(i) Key 3, followed by
(ii) Key 12, followed by
(iii) Key 9, followed by
(iv) Key 6

in the above order, highlighting clearly any splits or overflow.

4. Consider the following B^+-Tree.



Show each of the tree structure after the deletion of

(i) Key 5, followed by
(ii) Key 12, followed by
(iii) Key 9

in the above order, highlighting clearly any merging, underflow or redistribution.

5. Consider a generalization of the B-tree with minimum fullness factor $f = ¾$.

Determine:

(i) the mean storage utilization E(ρ)
(ii) the standard deviation of the storage utilization sd(ρ)
(iii) the probability that the storage utilization lies between 80% and 90%; i.e. Prob $[0.8 \leq \rho \leq 0.9]$.
(iv) the median of the storage utilization and compare it with the mean
(v) a general expression of the median for a tree with arbitrary minimum fullness factor *f*.

6. Consider a PARTS file with Part# as hash key, which includes records with the following Part# values:

{2305, 1168, 2580, 4871, 5659, 1821, 1074, 7115, 1620, 2428, 3943, 4750, 6975, 4981, 9208}

(i) Static Hashing – assume there is a fixed number of home buckets together with some overflow buckets.
The file uses 8 buckets, numbered 0 to 7. Each bucket is one disk block and holds a maximum of two records. Suppose we load these 15 records into the file in the given order using the hash function $h(K) = K \bmod 8$. Determine the average number of block accesses for a random record retrieval on Part#.

(ii) Extendible Hashing – with each bucket also holding a maximum of two records. Suppose now we load them into an expandable hash file based on extendible hashing, with hashing function $h(K) = K \bmod 128$. Show the resultant structure, clearly indicating the local and global depths, after the insertion of the 6^{th} record, assuming insertion is carried out in the order given.