Deadline: 28-SEP-2020 (Monday), 3:00 pm (late submission will not be accepted)

Points to note:

- Different books may have slightly different descriptions of concepts, algorithms and terminologies. To ensure fair assessment and uniformity in marking, you **must** follow the convention used in the lecture slides or our textbook (Database System Concepts). Other conventions will **not** be accepted.
- Students are expected to generalize the concepts they have learnt during the lecture in order to finish the assignment.
- You must show the steps clearly. The marker will not give you marks if s/he cannot understand your work.
- This is an **individual** assignment. You must work on your own. Check http://www6.cityu.edu.hk/ah/plagiarism.htm for "The Problem of Plagiarism".
- Submit the file to Canvas on or before the deadline**.**
- The file type must be either .docx file or .pdf file.
- Use your student ID(s) to name the file, such as 5xxxxxxx.docx or 5xxxxxxx.pdf.

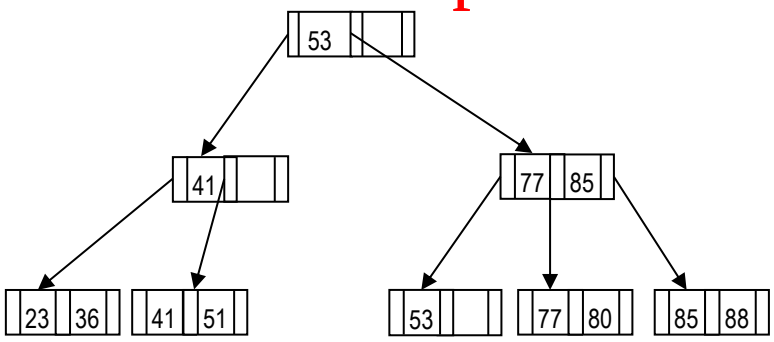
1. 16%

Regarding a B+-tree with parameter n, if there are K search-key values in the file, prove the following bounds for the tree height h (number of levels), assuming that $n > 3$ and $K > 1$.

$$\lceil \log_n(K) \rceil \le h \le \lceil \log_{\lceil n/2 \rceil}(K) \rceil$$

2. 42%

Given the following B+-tree with $n = 3$:



(a) Can you re-build a taller B+-tree with the **same** value of n using the **same** set of search-key values in the leaf nodes of the given tree? If yes, show the steps by drawing a new diagram whenever the height of the tree increases.

(b) Insert the search-key values 30, 83 and 10 in sequence to the given B+-tree and draw a new diagram for each insertion.

(c) Suggest a sequence of search-key values to be deleted from the resultant B+-tree in Part (b) to shrink the tree to **2** levels with the ***least*** number of deletions. Show the steps by drawing a new diagram whenever a node is deleted.

3. 42%

In extendable hashing, suppose each bucket can hold **three** records and the hash function h(K) = K mod 16 generates 4-bit values.

(a) At **most** how many records can be stored in the hash structure, without using any overflow buckets?

(b) At **least** how many records have to be inserted into an **empty** hash structure to make i = 4, where i is the length of the prefix of the hash value?

(c) Suggest a set of integral search-key values and an insertion sequence to illustrate your answer to Part (b). Draw a **new** diagram of the hash structure whenever the i value is incremented. Show the values of i and i_j clearly on the diagrams.

(d) Repeat Part (b) and Part (c), if empty buckets are **not** allowed during insertion.

(e) Consider a general case in which each bucket can hold n records and the hash function is h(K) = K mod 2^m. If empty buckets are **not** allowed, at **least** how many records have to be inserted into an **empty** hash structure to make $i = m$? Explain your answer.