Assignment 3

Due: Fri 27 May, 23:59

Question 1 (5 marks)

Answer the following questions and ***justify the reasons of your answers***.

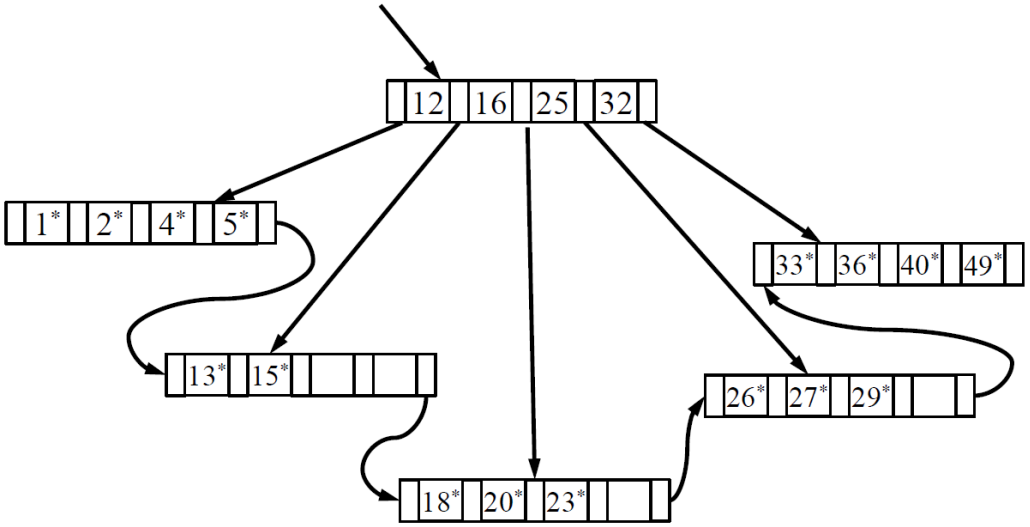
1) (1 mark) Suppose that a file is sorted against a search key A. An index is also built against the search key A. We also assume that A is a candidate key. Please show that using the index to conduct an equality search regarding A to find a record typically involves less I/O costs than that of conducting a binary search over the sorted file. Assume that the ordered file occupies *S* blocks and the index file occupies *C* blocks. (You can use a concrete index to justify your answers)

2) (2 marks) Is it possible that deleting an entry reduces global depth by 2 in the Extendible Hashing?

3) (2 marks) Find a scenario in Linear Hashing that splitting whenever an overflow occurs performs worse, in terms of the number of total pages, than splitting only when the number of overflow pages exceeds a given threshold.

Question 2 (5 marks)

Consider the B+-tree shown in the following as an original tree.



Answer the following questions.

1) (2 marks) Show the B+-tree after inserting a data entry with key 37 into the original tree.

2) (3 marks) Draw a valid B+-tree with the maximum number of tree nodes that contains the same data entries in the original tree. (The new B+-tree should have the same order as the original tree)

Question 3 (3 marks)

Consider a relation R(a,b,c,d,e) containing 6,000,000 records, where each data page of the relation holds 10 records. R is organized as a sorted file with the search key R.a. Assume that R.a is a candidate key of R, with values lying in the range 0 to 5,999,999. For the relational algebra $\pi_{a,b}\left(\sigma_{a>60,000}(R)\right)$, state which of the following approaches (or combination thereof) is most likely to be the cheapest:

1. Access the sorted file for R directly.
2. Use a clustered B+ tree index on attribute R.a.
3. Use a linear hashed index on attribute R.a.
4. Use a clustered B+ tree index on attributes (R.a,R.b).
5. Use a linear hashed index on attributes (R.a,R.b).
6. Use an unclustered B+ tree index on attribute R.b.

We assume that the database considers index-only plans. Index-only plans mean that an index contains all the columns needed to answer the query without having to access the data records in the files that contain the relations in the query.

Question 4 (6 marks)

An IT company developed a new database system to record the statics data of the coming Opera House Open Day including the number of reservations X, remaining gifts Y and meals ordered Z. Here is a schedule of three transactions:

S1, R1(X), S2, R2(Y), W1(X), E1, A, R2(X), S3, R3(X), B, W2(Y), E2, R3(Y), W3(X), W3(Z), E3

Si indicates the start point of transaction i;

Ei indicates the end point of transaction i;

Ri(X) indicates a read operation in transaction i on a variable X;

Wi(Y) indicates a write operation in transaction i on a variable Y;

Regarding the following questions, give and justify your answers.

1) (1 mark) Assume that the system crashes at B, what should be done to recover the system?

2) (1 mark) Assume a checkpoint is made at point A, what should be done to the three transactions when the crash happens at B?

3) (2 marks) Is the transaction schedule conflict serializable?

4) (2 marks) Construct a schedule (may not be the same as above) of these three transactions which causes deadlock when using two-phase locking protocol. If no such schedule exists, explain why.

Question 5* (1 mark)

In this big data age, many applications need to deal with the datasets that are too large to fit in memory. In these applications, I/O cost always dominates the overall execution costs and algorithms with less I/Os are always preferred. Assume that there is an online social networking website GC. It records all the user ids in a file. However, because of the mistakes made by the admin, the same user id may be stored multi-times in the file. The file has N user ids and the number of distinct user ids is K. Assume that the file is too large to fit in main memory. Devise an algorithm to remove the duplicates from the file. The output of the algorithm should be K distinct user ids among the N user ids in sorted order. The I/O complexity of your algorithm should be $O(\max\{\frac{N}{B} log_{\frac{M}{B}} \frac{N}{B} - \sum_{i=1}^{K} \frac{N_i}{B} log_{\frac{M}{B}} N_i, \frac{N}{B}\})$, where M is the capacity of main memory, B is block size and N_i is the number of copies of the i-th user id in the file. You need to give the I/O complexity analysis of your algorithm.

Note: questions labelled by * are difficult and will be marked strictly according to its correctness.

Assignment Submission

We accept electronic submissions only. Please submit your assignments as follows:

- Ensure that you are in the directory containing the file to be submitted. (note: we only accept files with .pdf extension)
- Type "give cs3311 ass3 ass3.pdf"

Note:

1. We do not accept e-mail submissions, and the submission system will be immediately closed after the deadline.
2. If the size of your pdf file is larger than 2MB, the system will not accept the submission. If you face this problem, try converting to compressed pdf.
3. If you have any problems in submissions, please email to longyuan@cse.unsw.edu.au.

Late Submission Penalty

Zero mark