

CS3402 Database Systems

Assignment 3

Question 1. (30 marks)

Suppose we have 10,000 STUDENT records of fixed length, and the disk has a block size of 2048 bytes. In this system, a block pointer is 4 bytes long ($P = 4$). Each STUDENT record has the following fields: Name (10 bytes), Age (4 bytes), Gender (4 bytes), Department (12 bytes), Student ID (12 bytes). The file is ordered by the key field Student ID, and we want to construct a primary index on Student ID. The records are stored unspanned.

- (1) Calculate the blocking factor (bfr) and the number of file blocks needed to store the STUDENT records. [10 marks]

Solution:

The record size $R = 10+4+4+12+12 = 42$ bytes

Then we could get the blocking factor (brf) by $\text{floor}(B/R) = \text{floor}(2048/42) = 48$ records/block

The number of blocks we need to store the STUDENT records is $I = \text{ceiling}(10000/48) = 209$

- (2) Assume that the primary index is a single-level index. Calculate the number of index entries and the number of index blocks. [10 marks]

Solution:

Each index entry has size $R_i = \text{size}(\text{Student ID}) + P = 12 + 4 = 16$ bytes.

The index blocking factor is $\text{bfri} = \text{floor}(2048/16) = 128$ entries/block

The number of index entry is 209.

Thus, the number of index blocks: $\text{ceil}(209/128) = 2$ blocks

- (3) Now suppose that we want to make the primary index a multilayer index. Calculate the number of hard disk blocks that will be occupied by this index file. [10 marks]

Solution:

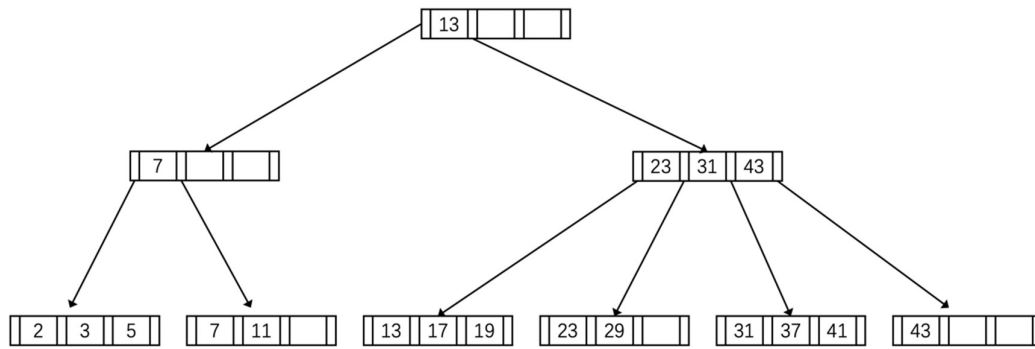
The fan-out for multilayer index is the same as the index blocking factor (bfri) which is 128 (see above).

The number of 1st level blocks L_1 is already calculated $L_1 = 2$ blocks. The number of 2nd level blocks is $L_2 = \text{ceil}(L_1/\text{fo}) = 1$

The total number of blocks required by the multilevel index file is $L_1+L_2 = 3$ blocks.

Question 2. (30 marks)

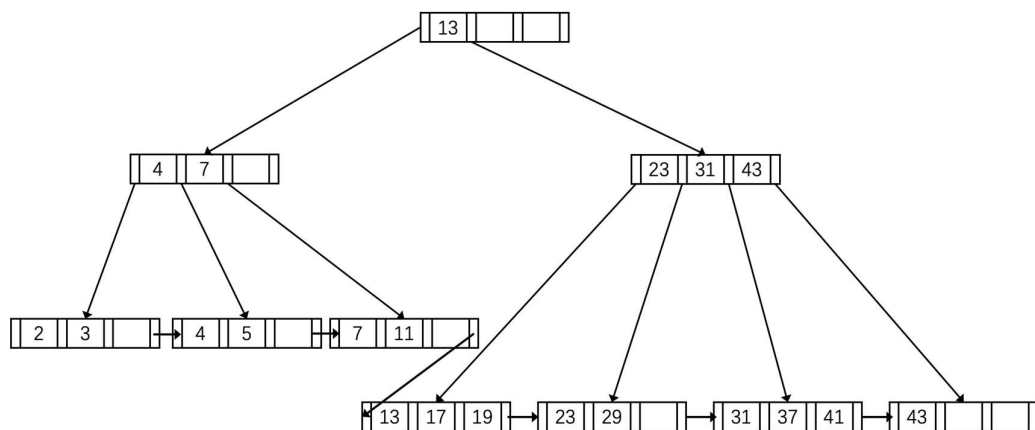
Consider the original B+ tree with order $d=4$ and height $h=3$ as follows.



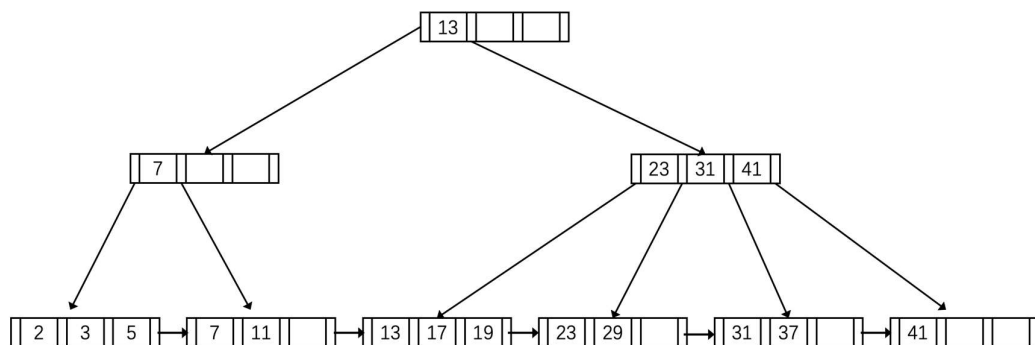
When answering the following questions, please follow the assumptions:

- A left pointer in an internal node guides towards keys smaller than its corresponding key, while a right pointer guided towards keys larger or equal than its corresponding key.
- A leaf or internal node underflows when the number of pointers goes below $\lceil \frac{d}{2} \rceil$.

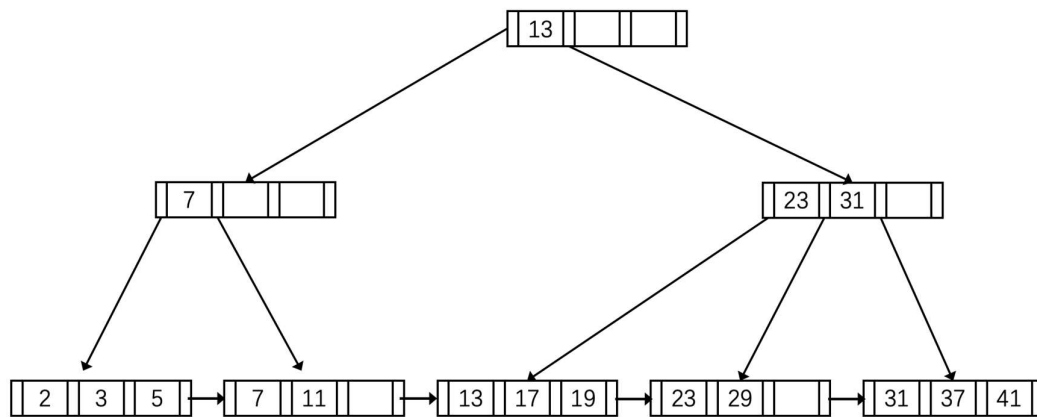
(1) Insert 4 into the original B+ tree. Draw the resulting tree. [15 marks]



(2) Delete 43 from the original B+ tree. Draw the resulting tree. [15 marks]



OR



Question 3. (40 marks)

Consider the three transactions T1, T2 and T3, and the schedules S1 and S2 given below.

T1: r1(x); r1(y); w1(x); c1

T2: r2(x); r2(z); w2(z); c2

T3: r3(z); r3(x); w3(x); c3

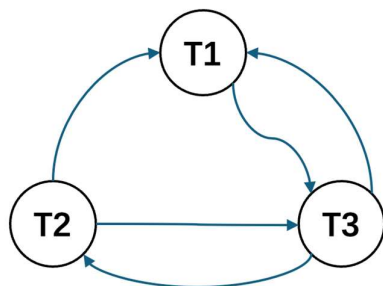
S1: r1(x); r1(y); r2(x); r2(z); r3(z); r3(x); w2(z); c2; w1(x); c1; w3(x); c3.

S2: r1(x); r1(y); r2(x); r2(z); w1(x); r3(z); r3(x); w3(x); c3; w2(z); c2; c1.

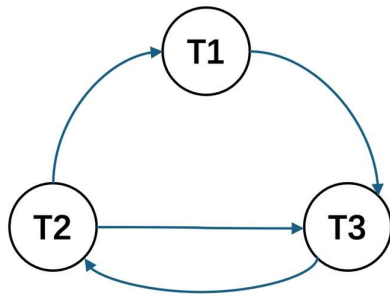
Notes: r1(x) represents transaction 1 read x; w2(y) represents transaction 2 write y; c3 represents transaction 3 commits.

- (1) Draw the serializability graphs for S1 and S2 and state whether each schedule is serializable or not.

S1: not serializable



S2: not serializable



(2) Determine whether the schedule is recoverable/nonrecoverable, cascadeless or strict.

S1: recoverable, cascadeless, strict

S2: nonrecoverable, so it is also not cascadeless and not strict.