Database Systems for Software Engineers

SOEN 363 - Winter 2023

Problem Set 3

Out: February 27, 2023

Due: March 19, 2023

1 B^+ Tree Basics [20 Points]

For the following sub-questions, consider the B^+ tree structure with order $d = 2$ (i.e. there are at most 4 keys per node, and at most 5 pointers to children) as shown in Figure 1 below.

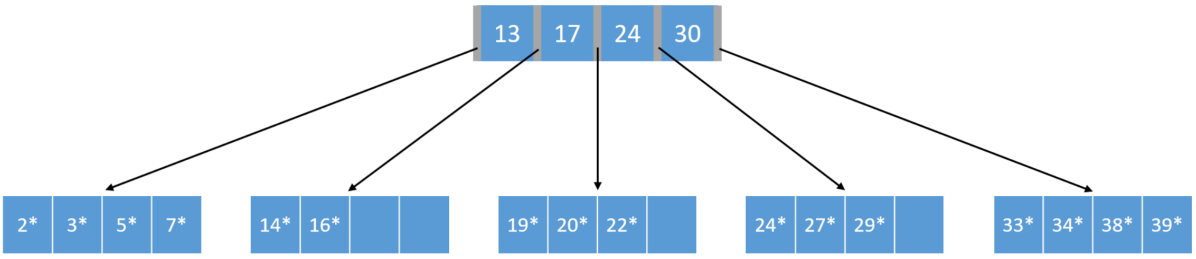
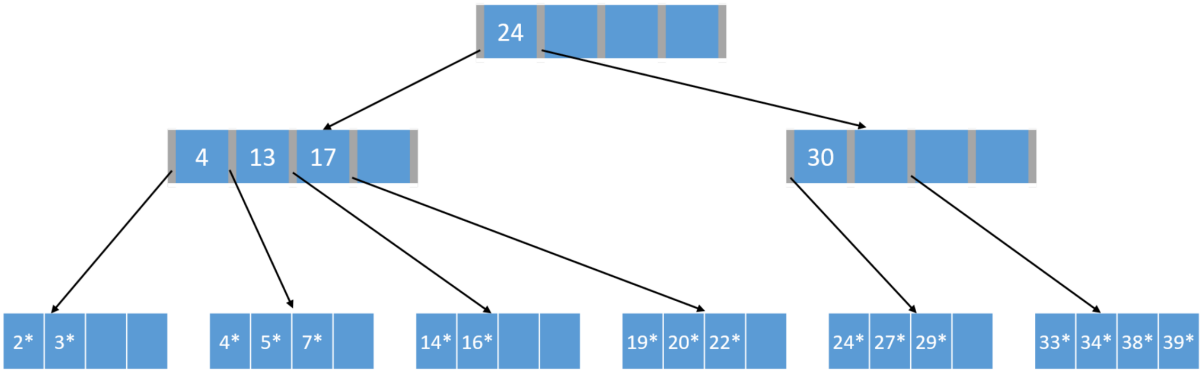


Figure 1: A B^+ tree with order $d = 2$

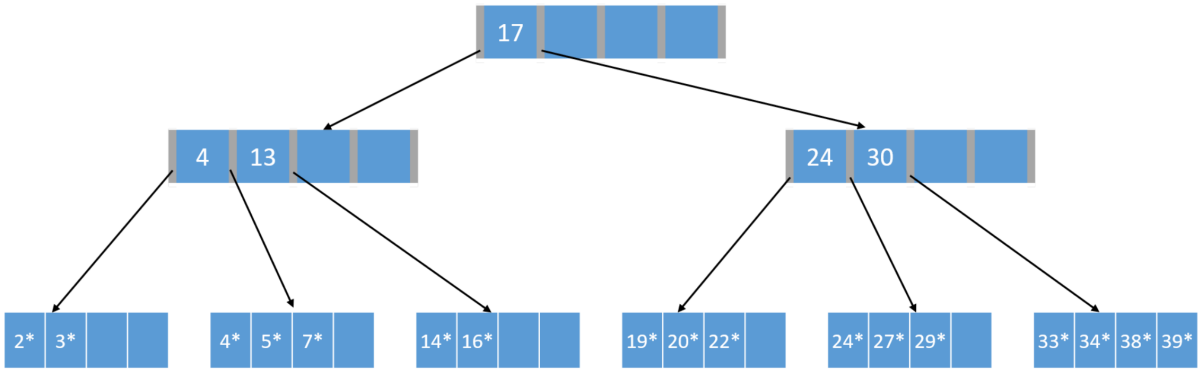
5pts (a) Assume the structure in Figure 1 is not a B^+ tree, but an **ISAM structure**. Show the new structure after inserting keys 10, 25 and 35.

Handout continues on the next page(s)

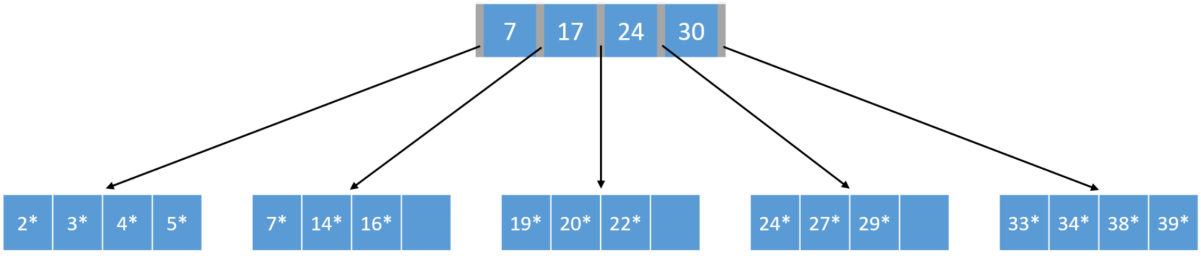
5pts (b) Starting from the original B^+ tree in Figure 1, we insert the record 4* with a key 4. From the 3 choices below, which is the resulting tree if we use the default "**redistribution with a sibling**" strategy: A, B, C, or neither? If neither, draw the correct tree.



Tree A

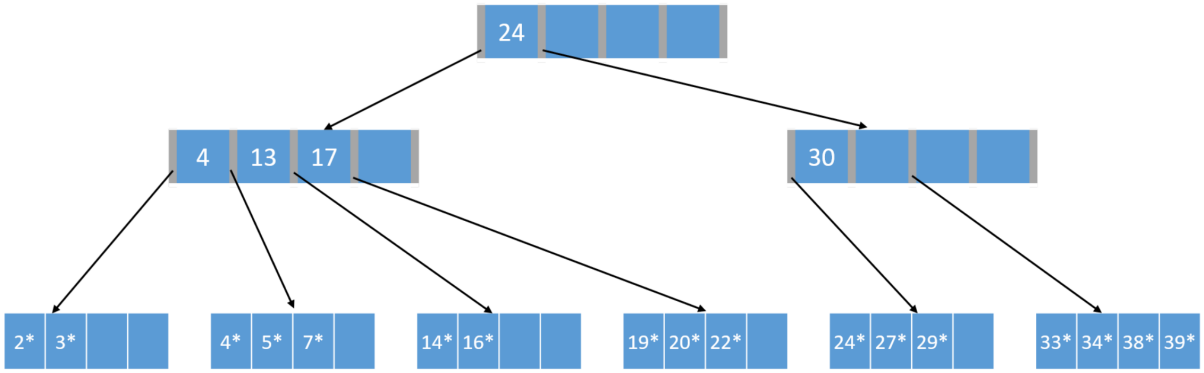


Tree B

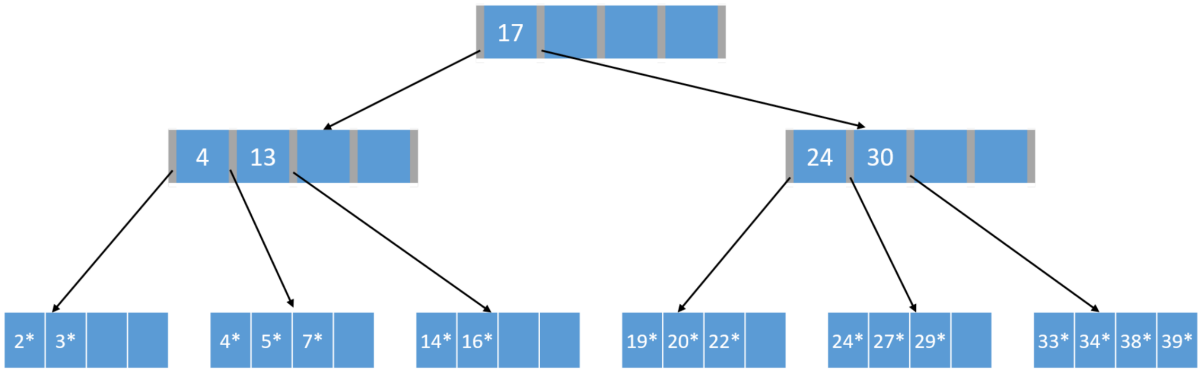


Tree C

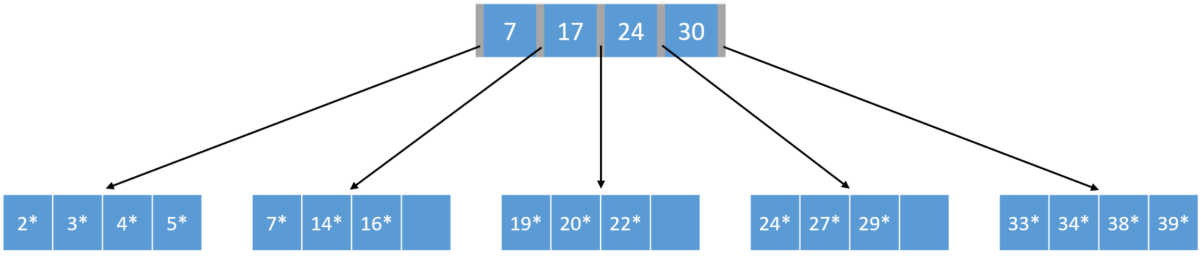
5pts (c) Starting from the original B^+ tree in Figure 1, we insert the record 4* with a key 4. From the 3 choices below, which is the resulting tree if we use the default "**no re-distribution**" strategy: A, B, C, or neither? If neither, draw the correct tree.



Tree A

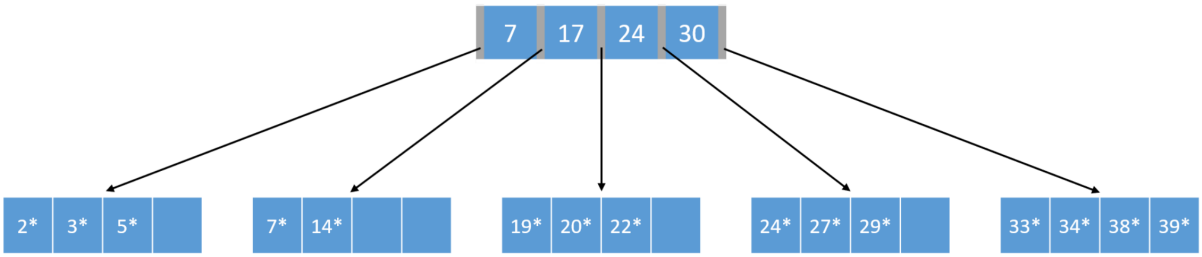


Tree B

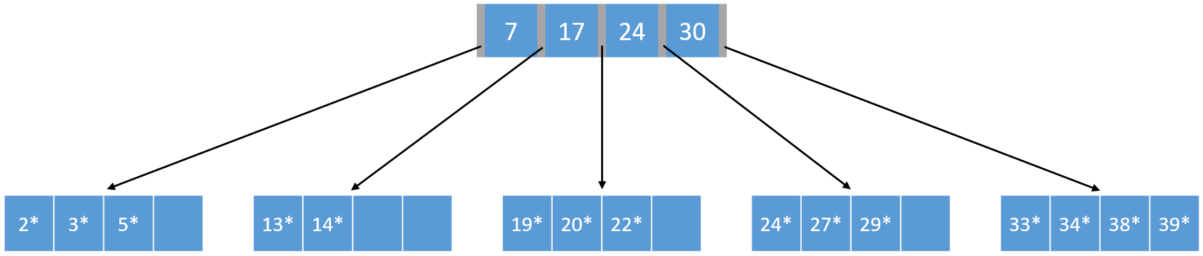


Tree C

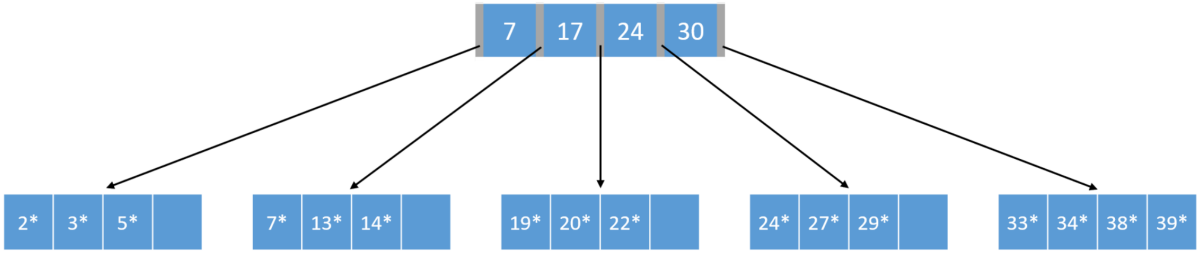
5pts (d) Starting from the original B^+ tree in Figure 1, we delete the record `16*`. From the 3 choices below, which is the resulting tree if we borrow a node from the right sibling: A, B, C, or neither? If neither, draw the correct tree.



Tree A



Tree B



Tree C

2 B^+ in Numbers [20 Points]

Assume that you have just built a dense B^+ tree index using **Alternative (2)** on a heap file containing **20,000** records. The key field for this B^+ tree index is a **50-byte** string, and it is a candidate key. Pointers (i.e., record ids and page ids) are (at most) **8-byte** values. The size of one disk page is **2000** bytes. The nodes at each level were filled up as much as possible.

10pts (a) How many levels does the resulting tree have? Show the steps performed to reach the result.

5pts (b) For each level of the tree, how many nodes are at that level? Show the steps performed to reach the result.

5pts (c) How many levels would the resulting tree have if key compression is used such that the average size of each key in an entry is reduced to 10 bytes and all pages are 70 percent full? Show the steps performed to reach the result.

Handout continues on the next page(s)

3 B^+ Trees (Clustered vs Unclustered) [30 Points]

Consider the instance of the Students relation (shown in Table 1) stored in file ***f***.

15pts (a) Construct a B^+ tree index of order **2** on the `gpa` field using **Alternative (3)**. The tuples in ***f*** are stored as: the first tuple is in *page 1, slot1*; the second is in *page 1, slot 2*; and so on. Each page can store up to four tuples. You may use (*page #, slot #*) to identify a tuple. Clearly indicate what the data entries are (i.e., *do not use the k* convention*).

15pts (b) Consider the following query:

```
SELECT sid, name FROM Students WHERE gpa >= 3.0 AND gpa <= 3.5
```

For each of the following, show the steps performed to reach the result when you calculate the IO cost (number of pages accessed) of finding all the tuples that fit the criteria when: (Note: Once a page is accessed all its contents is loaded to the memory, and no need to access it again)

6pts 1. tuples in ***f*** are sorted, and

6pts 2. tuples in ***f*** are unsorted (i.e., they appear in the order shown in table 1).

3pts 3. What do you conclude from this?

sid	name	login	age	gpa
53831	Maclayall	maclayan@music	11	1.8
53832	Guldu	guldu@music	12	3.8
53666	Jones	jones@cs	18	3.4
53901	Jones	jones@toy	18	3.4
53902	Jones	jones@physics	18	3.4
53903	Jones	jones@english	18	3.4
53904	Jones	jones(ggenetics	18	3.4
53905	Jones	jones@astro	18	3.4
53906	Jones	jones@chem	18	3.4
53902	Jones	jones@sanitation	18	3.8
53688	Smith	smith@ee	19	3.2
53650	Smith	smith@math	19	3.8
54001	Smith	smith@ee	19	3.5
54005	Smith	smith@cs	19	3.8
54009	Smith	smith@astro	19	2.2

Table 1: Instance of the students relation stored in file ***f***

4 Extendible Hashing [30 Points]

Assume we have the following records, where we indicate the hashed key in parentheses (in binary):

```
i        [001100]
h        [001100]
g        [101101]
f        [010010]
e        [111111]
d        [010010]
c        [100001]
b        [001100]
a        [000000]
```

Consider an Extendible Hashing structure where buckets can hold up to three records. Initially the structure is empty (only one empty bucket). Consider the result after the records above have been inserted in the order shown, using the lower-bits for the hash function. As mentioned in the textbook, assume that the directory doubles in size at each overflow.

5pts (a) Show the result after the above records are inserted.

2pts (b) What will be the global depth of the resulting directory?

3pts (c) How many buckets will we have?

4pts (d) List all the elements in the bucket which contains the element "i." What is the local depth of this bucket?

4pts (e) List all the elements in the bucket which contains the element "c." What is the local depth of this bucket?

4pts (f) Do we store the number of bits to use in the hash function in the **(a)** global or **(b)** local depth?

4pts (g) Consider the case that the directory just doubled. Is it true that every bucket will be split in two? (Yes/No)

4pts (h) If the local depth of a bucket is equal to the global depth of the directory, is this bucket pointed to by **(a)** exactly one, or **(b)** multiple directory entry(s)?

5 Submission

- The assignment is due at 11:59PM on March 19, 2023.
- The submission for this assignment consists of zipped file named as follows firstName_lastName_student ID. The zipped folder has to contain:
  - A PDF document with the answers of each question.
  - A README file where you specify any assumptions.
  - A signed "Concordia Expectation of Originality" form found here: `https://www.concordia.ca/content/dam/ginacody/docs/Expectations-of-Originality-Feb14-2012.pdf`
- If you have any problems with the submission, contact your respective TA:
  - Monday lab SA: Omij (o_manguk@live.concordia.ca)
  - Tuesday labs SB and SD: Philippe (p_arrie@live.concordia.ca)
  - Wednesday lab SC: Reham Omar (reham.omar@mail.concordia.ca)
  - Wednesday lab SE: Akshit (ak_desai@live.concordia.ca)

6 Late Policy

- If you hand in on time, there is no penalty.
- 0-24 hours late = 25% penalty.
- 24-48 hours late = 50% penalty.
- More than 48 hours late = you lose all the points for this project.