CS 411: Database Systems

Fall 2020

Homework 5 (Due by 23:59 CT on Nov 2)

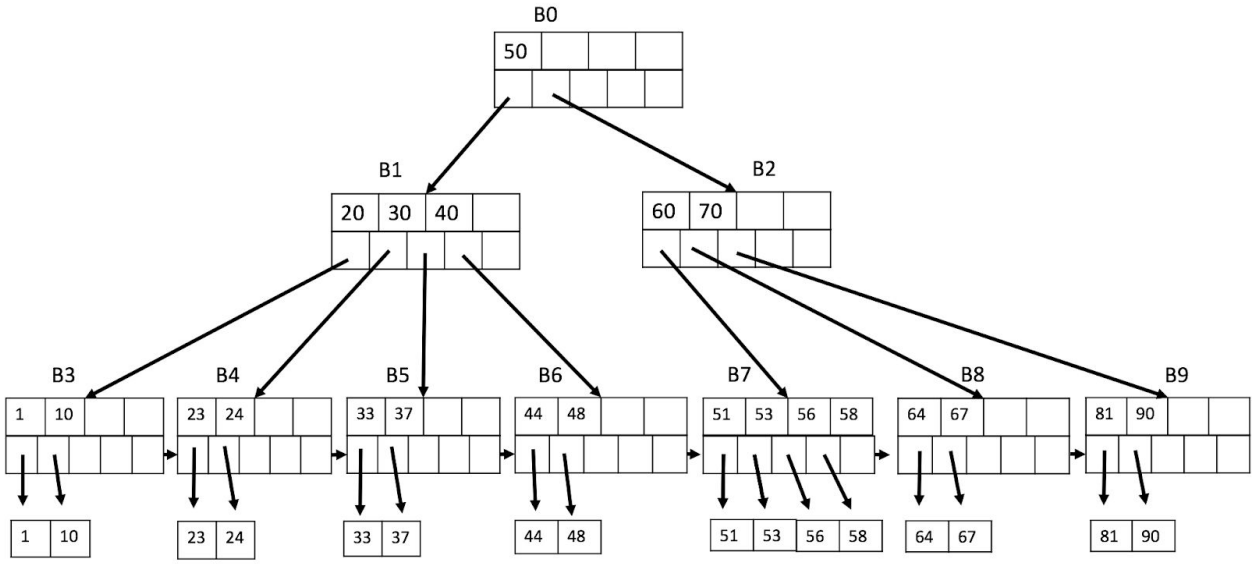
Logistics

1. The homework is due on 11/2 23:59 CDT. **We DO NOT accept late homework submissions.**
2. You will be using Gradescope to submit your solutions. Answer each sub-question (e.g. “A)”) on a **new page** and submit your solution as a single PDF file on Gradescope. The link to submission on **GradeScope** will be updated HERE soon. All registered students should have received access to GradeScope. Please submit the PDF to "Homework 5". *IMPORTANT*: Please make sure to link pdf pages with the corresponding question outline on GradeScope.
3. Please write all answers electronically. We won’t grade handwritten/hand-drawn versions. PowerPoint templates to draw the diagrams are provided with the release.
4. Keep your solutions brief and clear.
5. Please use Campuswire if you have questions about the homework but do not post answers. Feel free to use private posts or come to office hours.

Problem 1. B+ Tree Index (40 points)

Consider the following B+ tree with $d = 2$. For each part, execute the operation on the initial tree and answer the related question.

- For each of the lookup parts, describe which blocks would be read and in which order.
- For each of the insert and delete parts, apply all the operations to the **initial B+ tree** and draw the final resulting tree.
- For grading purposes, assume a cell prefers to merge with its **RIGHT** sibling instead of the left sibling when applicable. When a cell splits, assume the middle key goes to the **LEFT** resulting cell instead of the right one.



1. Lookup record with key 33 (3 points)
2. Lookup record with key 65 (3 points)
3. Lookup record with keys in the range 42 to 66 (4 points)
4. Insert record with key 55 (8 points)
5. Delete record with key 23 (8 points)
6. Delete records with key 56, 33, 81 in order, show your intermediate result after each deletion (14 points)

Problem 2. Extensible Hash Table (30 points)

Assume you have an extensible hash table with hash function h(k) = k mod 13, expressed as a binary string of size 4, and data block of size 2 (i.e., it can accommodate two tuples). You are asked to index the following key values in order: 34, 2, 29, 9.

1. Draw the extensible hash table which obeys the above constraints after the four keys are inserted. (20 points)

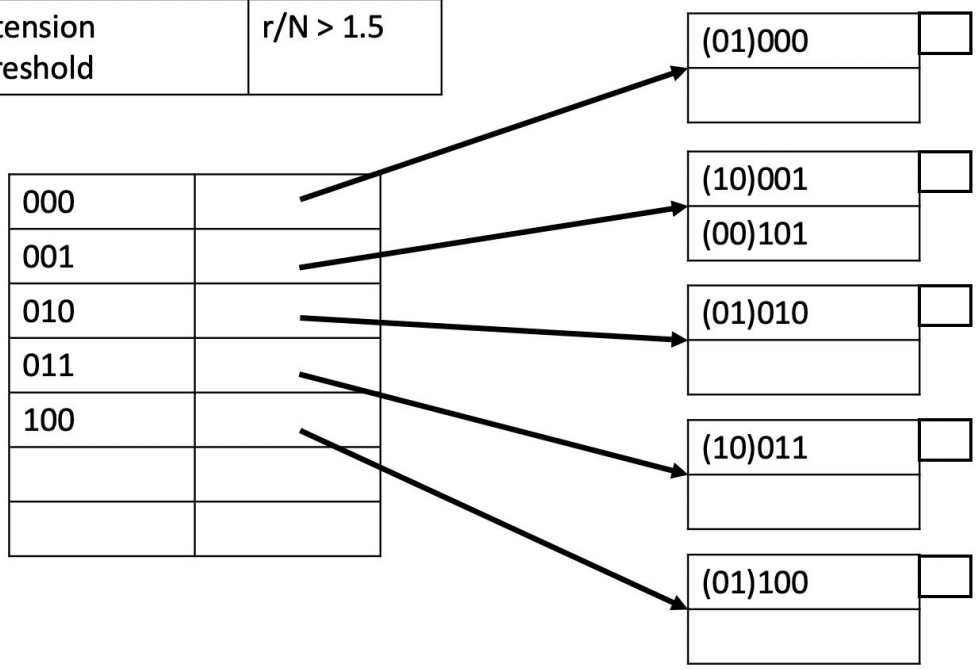
2. Using your solution to the **previous question**, now consider the insertion of keys 10 and 27 into the hash table, and draw the resulting hash table. (10 points)

Problem 3. Linear Hash Table (40 points)

Consider the following linear hash table. For each part, execute the operation on the **initial hash table** and answer the related question. **Draw the hash table after each insertion operation and update the upper left table** (statistics tab).

1. Calculate r/N at initial state. (5 points)
2. Insert key (00)111 into the hash table. (10 points)
3. Insert key (01)001 into the **initial** hash table, and then insert key (01)101. Note: Draw the intermediate hash table after the first insertion to earn partial credits. (25 points)

Number of buckets	N = 5
Number of bits	i = 3
Number of records	r = 6
Extension threshold	r/N > 1.5



Problem 4. Choosing a good index (20 points)

Consider the following table of all the Movies in our database.

```
CREATE TABLE Movie (
   MovieId int,
   Name varchar(255),
   Company varchar(255),
   Director varchar(255),
   Genre varchar(255),
   Ratings int,
   ReleaseYear int,
   PRIMARY KEY (MovieId)
);
```

With the following indexes:

INDEX1: Clustered hash table index on MovieId

INDEX2: Clustered B+ tree index on MovieId

INDEX3: Unclustered hash table index on Name

INDEX4: Unclustered B+ tree index on Name

INDEX5: Unclustered hash table index on ReleaseYear

INDEX6: Unclustered B+ tree index on ReleaseYear

INDEX7: Unclustered hash table index on Company

For each of the given predicates, describe which index is the best to pick and why it's the best? Note that you can choose no index and do a full scan on the table if you think it will outperform using any indexes.

1. SELECT * FROM Movie WHERE Name LIKE 'M%' (5 points)
2. SELECT * FROM Movie WHERE ReleaseYear = 1996 (5 points)
3. SELECT * FROM Movie WHERE MovieId > 100 and MovieId < 247 (5 points)
4. SELECT * FROM Movie WHERE Company <> 'Lionsgate Films' (5 points)

For part 4, 'Lionsgate Films' produces very few films compared to other movie companies in our database.