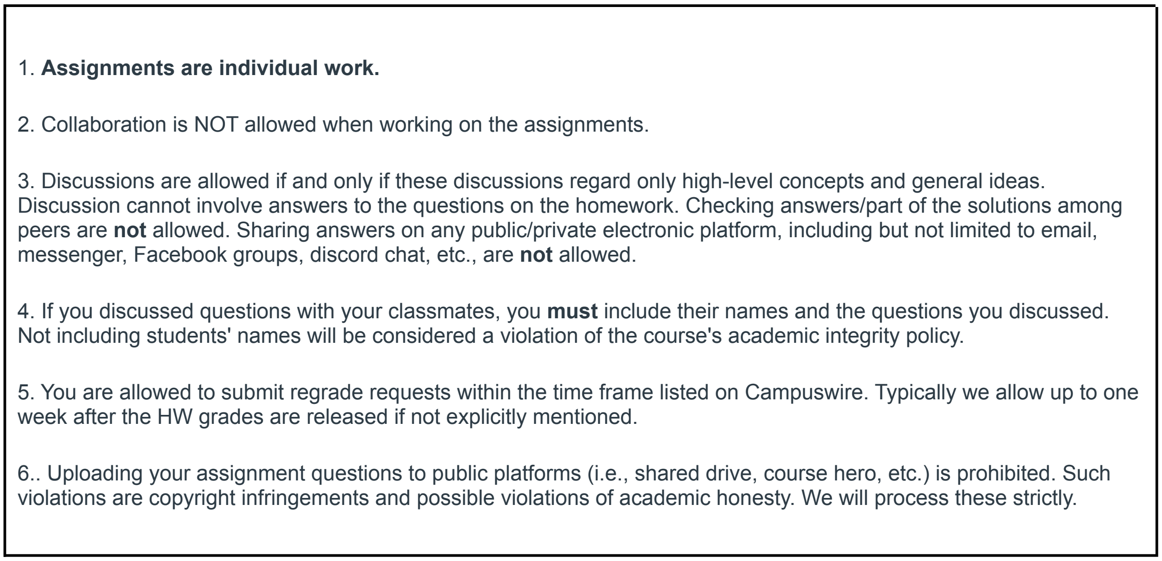
CS 411: Database Systems

Spring 2023

Homework 3 (Due by 23:59 CT on March 4, 2023)

Logistics

1. This homework is due on March 4th, 2023 at 23:59 CT. **We DO NOT accept late homework submissions.**
2. You will be using Gradescope to submit your solutions. Answer each sub-question (e.g. “a.”, “b.” etc.) on a **new page** and submit your solution as a single PDF file on Gradescope. All registered students should have received an email invitation to Gradescope. Please submit the PDF to "Homework 3".
3. ***IMPORTANT*: Please make sure to link PDF pages with the corresponding question outline on Gradescope.**
4. The answers can be written electronically or they can be hand-written, but if we cannot read your submissions, we won’t be able to grade them.
5. Please write down any intermediate steps to receive full credit.
6. Keep your solutions brief and clear.
7. 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.
8. DO NOT PLAGIARIZE. The following are written in the course syllabus:



1. **Assignments are individual work.**

2. Collaboration is NOT allowed when working on the assignments.

3. Discussions are allowed if and only if these discussions regard only high-level concepts and general ideas. Discussion cannot involve answers to the questions on the homework. Checking answers/part of the solutions among peers are **not** allowed. Sharing answers on any public/private electronic platform, including but not limited to email, messenger, Facebook groups, discord chat, etc., are **not** allowed.

4. If you discussed questions with your classmates, you **must** include their names and the questions you discussed. Not including students' names will be considered a violation of the course's academic integrity policy.

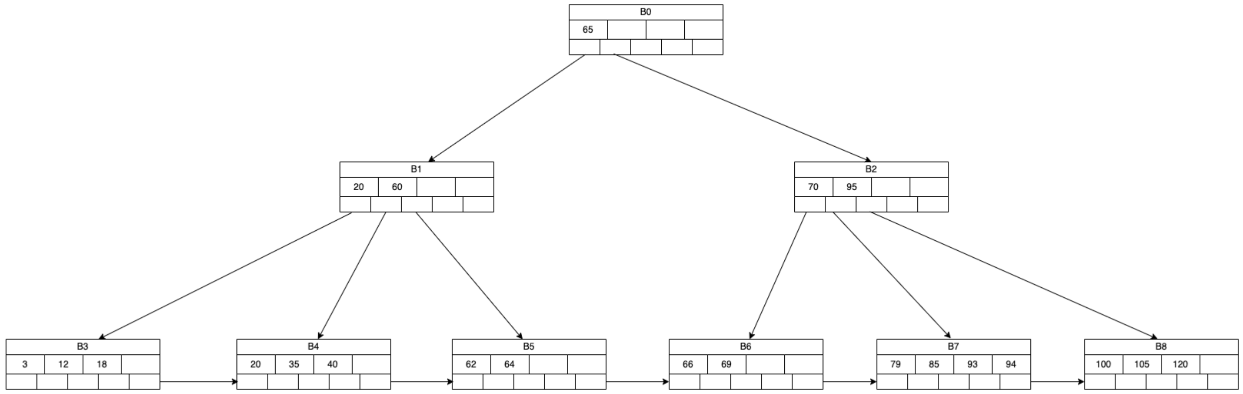
5. You are allowed to submit regrade requests within the time frame listed on Campuswire. Typically we allow up to one week after the HW grades are released if not explicitly mentioned.

6.. Uploading your assignment questions to public platforms (i.e., shared drive, course hero, etc.) is prohibited. Such violations are copyright infringements and possible violations of academic honesty. We will process these strictly.

Section 1 B+ Tree

Problem 1: Searching a B+ Tree

Consider the following B+ tree with $d = 2$. All the numbers in the B+ Tree represent IDs.



1.1 (1 point) Please write down the blocks visited for the following query:

```
SELECT * FROM table WHERE ID = 12
```

1.2 (1 point) Please write down the blocks visited for the following query:

```
SELECT * FROM table WHERE ID <= 41
```

1.3 (1 point) Please write down the blocks visited for the following query:

```
SELECT * FROM table WHERE ID > 68
```

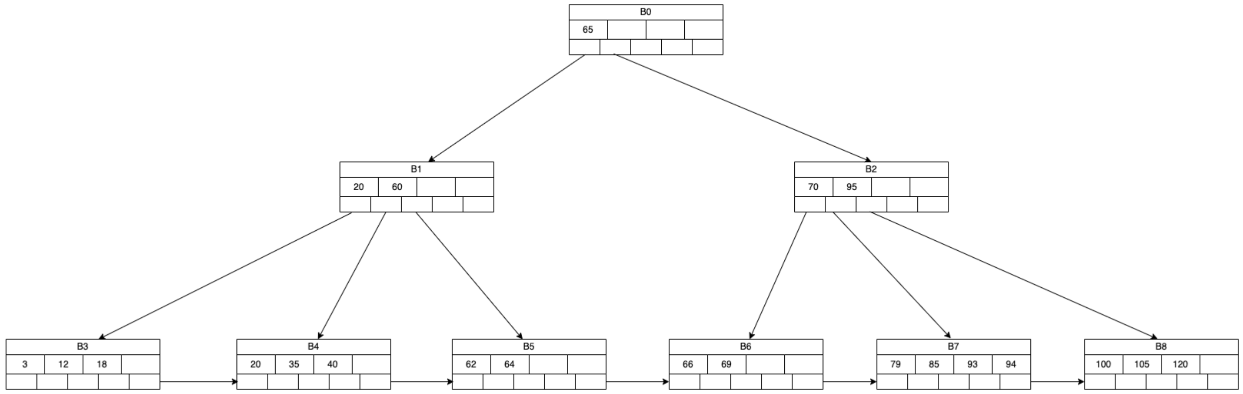
1.4 (2 point) Please write down the blocks visited for the following query:

```
SELECT * FROM table WHERE ID > 80 AND ID < 120
```

Problem 2: Insertion in a B+ Tree

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

- All the numbers in the B+ Tree represent IDs.
- Each sub-question is independent. For each of the insertions, apply all the operations to the **initial B+ tree** and draw the final resulting tree.
- For grading purposes, when a cell splits, assume the middle key goes to the **LEFT** resulting cell instead of the right one. If a rotation is possible with either sibling, assume a cell prefers to borrow from its RIGHT sibling.
- When updating values in the parent node, please reuse values already existing in the leaf node.
- Highlight all changes made to the tree in each step
- You can duplicate the B+ Tree draw.io file from draw.io file / png file



2.1 (1 point) Insert 19 into the tree, you must show all intermediate steps.

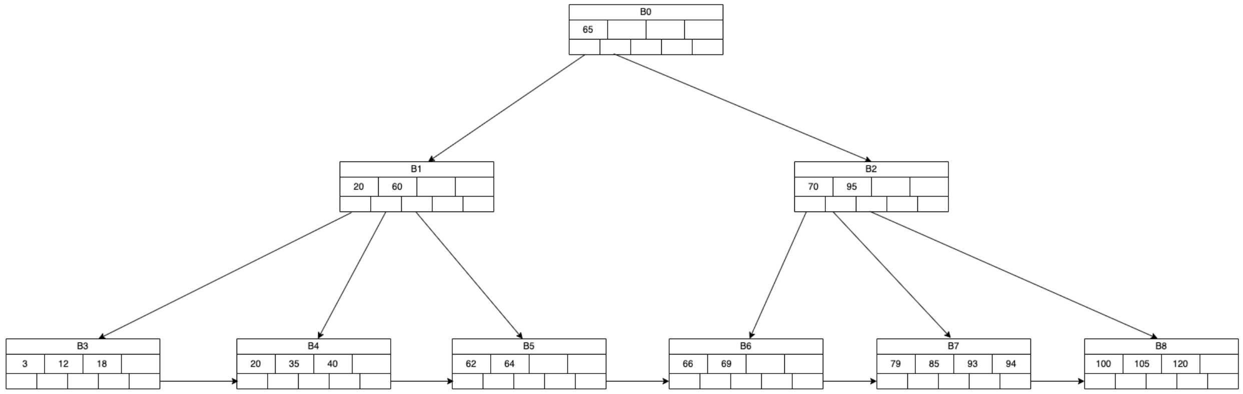
2.2 (4 point) Insert 44 and 55 sequentially into the tree, you must show all intermediate steps.

2.3 (5 point) Insert 67, 130 and 89 sequentially into the tree, you must show all intermediate steps.

Problem 3: Deletion from a B+ Tree

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

- All the numbers in the B+ Tree represent IDs.
- Each sub-question is independent. For each of the deletions, 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. If a rotation is possible with either sibling, assume a cell prefers to borrow from its RIGHT sibling.
- When updating values in the parent node, please reuse values already existing in the leaf node.
- Highlight all changes made to the tree in each step
- You can duplicate the B+ Tree draw.io file from draw.io file / png file



3.1 (1 point) Remove 40 from the tree, you must show all intermediate steps.

3.2 (3 point) Remove 69 from the tree, you must show all intermediate steps.

3.3 (6 point) Remove 18, 35 and 40 sequentially from the tree, you must show all intermediate steps.

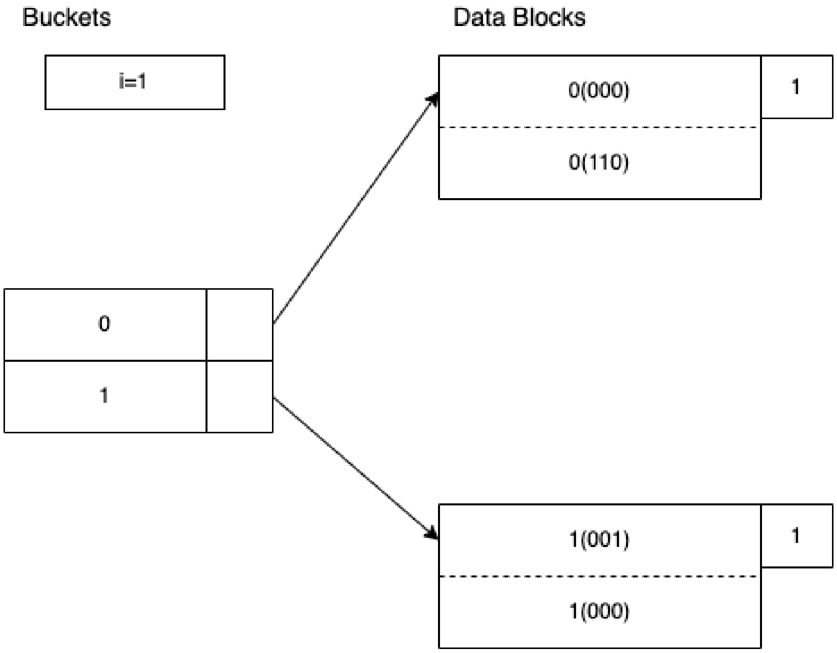
Section 2 Hash tables

Problem 4: Extensible Hash Table (10 points)

Consider the following extensible hash table. Please insert the following values into the hash table in sequence: 0001, 0111, 1110, 0101. Please draw all intermediate steps.Highlight all changes made to the buckets/data blocks in each step

Note: The buckets can be drawn in any order for the solution.

You can duplicate the EH draw.io file from here: draw.io file / png file

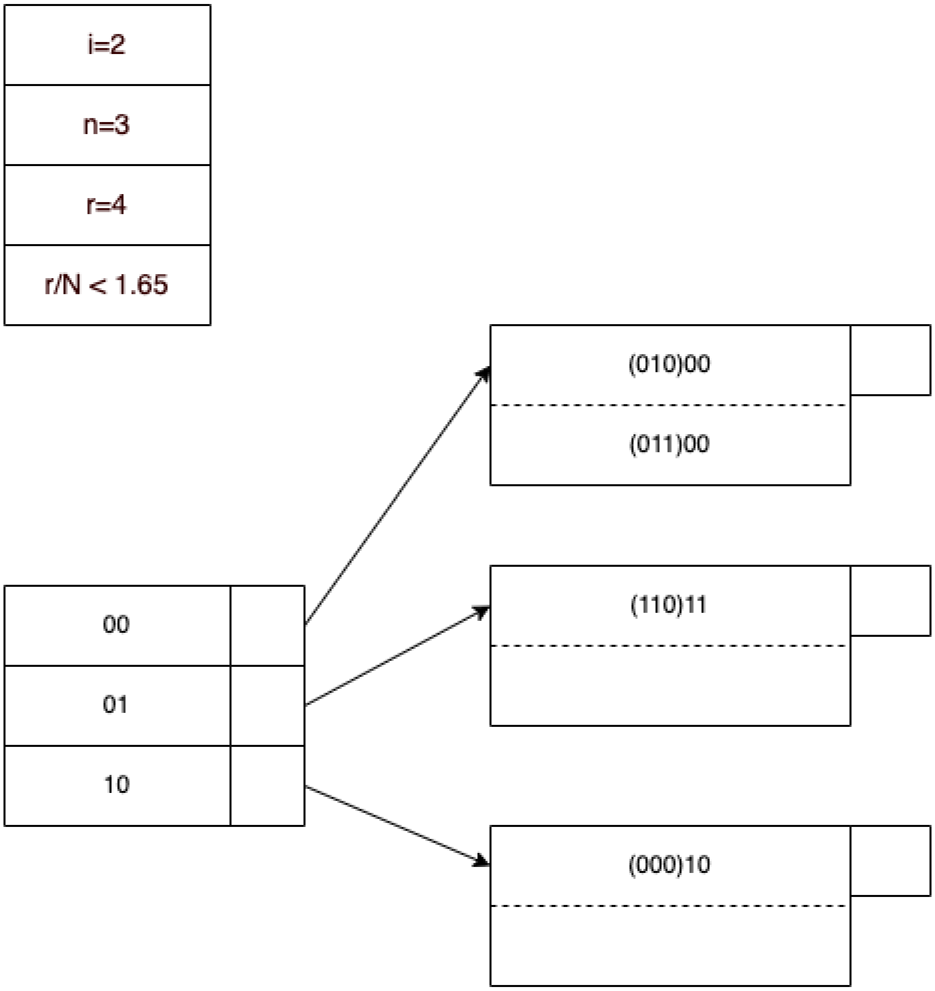


Problem 5: Linear Hash Table (10 points)

Consider the following Linear hash table. Please insert the following values into the hash table in sequence: 11000, 00101, 00001, 01111. Please draw all intermediate steps and don’t forget to denote bit flips.Highlight all changes made to the buckets/data blocks in each step

Note: The buckets can be drawn in any order for the solution

You can duplicate the LH draw.io file from here : draw.io file / png file



i=2
n=3
r=4
r/N < 1.65
00
01
10
(010)00
(011)00
(110)11
(000)10