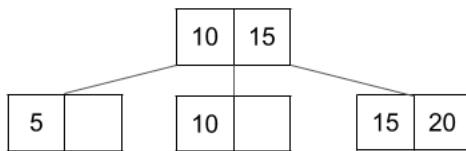# CS 186 - Fall 2020
Exam Prep Section 2
B+ Trees, Sort/Hash, Buffer Management, and Relational Algebra

Sunday, September 27, 2020

1   B+ Trees

Given the following (degree $d = 1$) B+ tree:

(a) Draw what the tree looks like after adding 2, 6, and 12 in that order.

For the following questions, consider the tree directly after 2, 6, and 12 have been added.

(b) What is the maximum number of inserts we can do without changing the height of the tree?

(c) What is the minimum number of inserts we can do that will change the height of the tree?

2 More B+ Trees

Consider the following schema:

```
CREATE TABLE FineWines (
    name CHAR(20) NOT NULL,
    years_aged INTEGER NOT NULL,
    price INTEGER NOT NULL
);
```

Suppose that we have built an index over `<years_aged, price>`, and suppose this index is two levels deep (in addition to the root node), and data is stored by *list of references* in separate data pages, each of which can hold hundreds of records.

(a)
```
SELECT * FROM FineWines
WHERE years_aged = 5
   AND price = 100
```

What is the worst case number of page I/Os to execute this query on an unclustered index if there is 1 matching record? 2 matching records? 3 matching records?

(b) What is the worst case number of page accesses to execute this query on a clustered index if there is 1 matching record? 2 matching records? 3 matching records?

(c)
```
SELECT * FROM FineWines
WHERE price = 100
```

What is the worst case number of page I/Os to execute this query if there are 150 data pages?

(d)
```
DELETE FROM FineWines
WHERE years_aged = 1
   AND price = 5
```

Suppose this query deletes exactly one record. How many page I/Os are needed to execute this query? (Assume no tree rebalancing is required.)

3   Sorting and Hashing

Suppose the size of a page is 4KB, and the size of the memory buffer is 1 MB (1024 KB).

1. We have a relation of size 800 KB. How many page IOs are required to sort this relation?

2. We have a relation of size 5000 KB. How many page IOs are required to sort this relation?

3. What is the size of the largest relation that would need two passes to sort?

4. What is the size of the largest relation we can possibly hash in two passes (i.e. with just one partitioning phase)?

5. Suppose we have a relation of size 3000 KB. We are executing a `DISTINCT` query on a column `age`, which has only two distinct values, evenly distributed. Would sorting or hashing be better here, and why?

6. Now suppose we were executing a `GROUP BY` on `age` instead. Would sorting or hashing be better here, and why?

4 External Sorting

Assume our buffer pool has 8 frames. In this question, we'll externally sort a 500 page file.

1. How many passes will it take to sort this file?

2. Given the number of passes you calculated in 4.1, how many I/Os are necessary to externally sort the file?

3. What is the minimum number of additional frames needed to reduce the number of passes found in 4.1 by 1?

4. What is the minimum number of additional frames needed to sort the file in one pass?

5 Hashing

1. Suppose the size of each page is 4KB, and the size of our memory buffer is 64KB. What would be the I/O cost of hashing a file of 128 pages, assuming that the first hash function creates 2 partitions of 32 pages, and all other partitions are uniformly partitioned?

6 Buffer Management

We're given a buffer pool with 4 pages, which is empty to begin with. Answer the following questions given this access pattern:

$$A \ B \ C \ D \ E \ B \ A \ D \ C \ A \ E \ C$$

1. What is the hit rate for MRU?

2. In order of when they were first placed into the buffer pool, what pages remain in the buffer pool after this sequence of accesses?

3. What is the hit rate for clock?

4. Which pages are in the buffer pool after this sequences of accesses?

5. Which pages have their reference bits set?

6. Which page is the arm of the clock pointing to?

7   Relational Algebra

1. Consider two relations with the same schema: R(A,B) and S(A,B). Which one of the following relational algebra expressions is not equivalent to the others?

   (A) $\pi_{\mathrm{R.A}}((R \cup S) - S)$
   (B) $\pi_{\mathrm{R.A}}(R) - \pi_{\mathrm{R.A}}(R \cap S)$
   (C) $\pi_{\mathrm{R.A}}((R - S) \cap \pi_{\mathrm{R.A}}(R)$
   (D) They are all equivalent.

2. The CS 186 TAs have decided to go into real estate! All of the information we need is described by the following schema, where the primary key of each table is underlined:

   ```
   Homes(home_id int, city text, bedrooms int, bathrooms int, area int)
   Transactions(home_id int, buyer_id int, seller_id int,
                transaction_date date, sale_price int)
   Buyers(buyer_id int, name text)
   Sellers(seller_id int, name text)
   ```

   Write the relational algebra plan to find the duplicate-free set of id's of all homes in Berkeley with at least 6 bedrooms and at least 2 bathrooms that were bought by "Bobby Tables".

8   Hints for Question 3: Sorting and Hashing

1. Convert all numbers to pages, and then apply the equation you learned in class.

2. Same.

3. Again, convert all numbers to pages.
   The second pass merges how many runs?
   For the second pass to complete the sort, this must be enough to merge all the runs created by the first sort. How big does the relation have to be for the first sort to create that many runs?

4. Similar to the previous question, but think about the hashing protocol instead.

5. Which protocol benefits from removing duplicates?

6. `GROUP BY` does not allow us to remove duplicates. Are both protocols capable of handling such large numbers of duplicates?