Database Systems, CSCI 4380-01
Homework # 7
Due Friday 16, 2018 at 11:59 PM

Introduction.

This homework is worth 4% of your total grade. If you choose to skip it, the final exam will be worth 4% more. If you complete all the homeworks after Exam #2, there will be a special bonus in terms of your final grade computation.

This homework is on learning how B-trees are used to answer different types of queries. It requires you to write code to solve it. You are free to use any programming language you want. As you will see in the deliverables, we will ask you to submit both your program output and your code. We will mostly check your output for grading but we reserve the right to execute your program if we feel the need to do so. Submit only working code and its output. Please follow the shown output format for easily checkable results.

Due to its size, the data for this homework is not on Piazza. You can download it here:

http://www.cs.rpi.edu/ sibel/DBS_Past_Materials/Fall2018.

Data Format

In this homework, you will use a new data format that mimics have tables and indices are stored on disk. You are given multiple folders. Each one contains the same data, tuples in the `clues` table, sorted differently on disk. There is also pages for a single index in each folder, the index is on `clues(gameid, clueid, category)`.

Data pages. The tuples in the `clues` table are stored in different files named `page#.txt`, representing disk pages. Each such file contains a number of tuples, one each line, fields are separated by `|`. The fields are ordered as follows:

```
gameid, clueid, clue,value,category, cat_type, isdd, correct_answer
```

Index pages. Each index page correspoding to a specific node is also stored in a separate file, named `index#.txt` except for the root that is named `index_root.txt`.

Each index page has a header that lists whether it is it a `Leaf` or an `Internal` node. Leaf nodes have a sibling pointer, the name of the file containing the next leaf node, e.g. `Leaf|index6.txt`. The last leaf node has no sibling pointer and will have the value `Leaf|-`.

Each line of an index page contains an entry and a pointer. The entry is the value of the indexed attributes.

Pages containing leaf nodes point to the page containing the indexed tuple:

```
1110|4|MY NAME IS URL|page222.txt
```

means that a tuple for `gameid=1100, clueid=4, category='MY NAME IS URL'` is stored in data page `page222.txt`.

Pages containing internal nodes contain pointers to other index pages:

```
2897|5|IT AIN'T BRAIN SURGERY|index418.txt
```

means that the leaf node `index418.txt` points to entries with the smallest value `gameid=2897, clueid=5, category='IT AIN''T BRAIN SURGERY'`.

Remember, index is on `clues(gameid, clueid, category)`, which means the tuples are sorted by gameid first, then by clueid next and then by category. Given that gameid and clueid is unique in this relation, sorting by category does not have an effect. It is there as additional information.

Alternate Folders. You are given three separate folders, `data1, data2, data3`. Each one contains the same information but the underlying table is stored with respect to a different order. This will impact the query costs. For each query given to you, you must assess the query costs for each folder separately and report.

Problem Specification

Your job in this homework is to use the index to answer a query and then output the cost of this query based on the data in each of the three folders. You do not need to return the answer to the query and you can assume the number of levels of the B-tree is fixed.

The cost of a query is given by the number of disk pages (i.e. files) that must be read to answer this query.

Each query is answered by scanning the index first, starting with root, an internal node and the first leaf node. If the range spans multiple leaf nodes, then you will use the sibling pointers to scan towards right until a value outside of the range is found.

For each found tuple, you will read the data pages if necessary (i.e. the query requests an attribute that is not in the index). If the same data page is found for multiple tuples, you can assume you read a data page only once.

For each query, print out number of tuples found, number of index pages (including root) and data pages read.

Query Syntax. Each query specifies a value or range for three attributes `gameid, clueid, category` and a list of attributes to return. If there is no upper/lower range for a condition, it is listed as empty using Python slicing syntax.

For example: for `gameid`:

- `[10:10]`: same as `gameid=10`.
- `[:20]`: same as `gameid<=20`.
- `[30:]`: same as `30<=gameid`.
- `[40:50]`: same as `40<=gameid and gameid<=50`.

- `[:]`: means all `gameid` values are allowed.

So, the following query:

```
[10:10]|[3:4]|[:]|gameid,clueid,clue
```

is equivalent to:

```
select gameid, clueid, clue from clues where gameid=10 and 3<=clueid and clueid<=4
```

```
[:]|[:]|[A:B]|category,clue
```

is equivalent to:

```
select category, clue from clues where 'A' <= category and category <= 'B'
```

Submission Instructions.

You will submit two files to SUBMITTY: your program output as a text file and your program source code.

You will be a given an input file with different test conditions before the deadline. Each line of the file will contain a query of the above form. Run this against your code and submit your output. Please math our format because we will use diff to test its correctness.

In your output file, you will list the query on one line, followed by three lines, one for the query cost for each folder, followed by a single new line.

Name this text file `username_hw7ans.txt` and submit it.

Suppose you have the following input file:

```
2140:2140|:|:|category
2140:2140|:|:|clue,category
```

Here is how the output file will look (correctness is not yet guaranteed, but I will post a few answers later in the week):

```
Query: 2140:2140|:|:|category
Folder:hw7_data/data1 Index Pages: 3 Data Pages: 0
Folder:hw7_data/data2 Index Pages: 3 Data Pages: 0
Folder:hw7_data/data3 Index Pages: 3 Data Pages: 0

Query: 2140:2140|:|:|clue,category
Folder:hw7_data/data1 Index Pages: 3 Data Pages: 1
Folder:hw7_data/data2 Index Pages: 3 Data Pages: 4
Folder:hw7_data/data3 Index Pages: 3 Data Pages: 4
```