Carnegie Mellon University
Department of Computer Science
15-826 Multimedia Databases and Data Mining
C. Faloutsos, Fall 2019

# Homework 1 - Solutions

Due: **hard copy, in class, at 1:30pm, on 09/16/2019**

**VERY IMPORTANT - check-list**:

1. Deposit **hard copy** of your answers, in class. For ease of grading, please **type** the full info on each page:
   - your *name* and *Andrew ID*,
   - *Course#* and *Homework#*.
2. **Typeset** all of your answers (eg., ascii, pdf, msword, etc). Handwritten responses may get **zero** points, at the discretion of the grader.
3. **Staple** them, if you use more than 1 page.

**Reminders:**

- *Plagiarism*: Homework is to be completed *individually*.
- *Late homeworks*: please follow standard policy, i.e., please email your homework
  - to the TA
  - with the subject line exactly `15-826 Homework Submission (HW 1)`
  - and the count of slip-days you are using.

*For your information:*

- Graded out of **100** points; **3** questions total
- Rough time estimate: *2-6 hours*

*Revision* : 2019/09/25 15:48

| Question | Points | Score |
|---|---|---|
| B-trees | 5 | |
| Linear Hashing | 5 | |
| SQL | 90 | |
| Total: | 100 | |

## Question 1: B-trees . . . . . [5 points]

Consider B-trees of order $d$=2 (2*$d$+1 = 5 = maximum fanout). One such tree is in Figure 1.

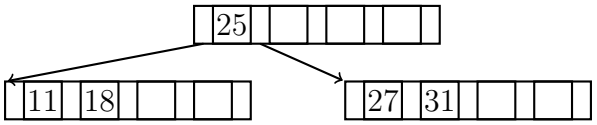


Figure 1: A B-tree of order $d$=2, with $n$=3 nodes, and height $h$=2.

*NO NEED* to justify your answers.

(a) **[2 points]** In an initially empty B-tree of order 2, insert the first 4 integers: 1,2,3,4. How many nodes will the tree have?

(a) **1**

(b) **[3 points]** In an initially empty B-tree of order 2, insert the first 10 integers: 1,2,. . .,10. How many nodes will the tree have?

(b) **4**

## Question 2: Linear Hashing . . . . . [5 points]

Consider a hash table that operates under linear hashing. When it started, the initial hashing function was $h_0(x) = x \bmod 7$; the hash table had $B$=7 buckets (0,1,2,. . .,6), and the split pointer was $s$=0.

After some other splits, the table has $B'$=28 buckets (numbered 0,1,. . . , 27). Answer the following questions. *NO NEED* to justify your answers.

(a) **[2 points]** Where is the split pointer $s$? Give the bucket number it points to (integer in the range 0, . . . , 27)

(a) **0**

(b) **[3 points]** How many hashing functions are active?

(b) **1**

## Question 3: SQL . . . . . [90 points]

For this part, we will use `sqlite3` (version 3.7.17), which is available on the andrew unix machines (`ssh unix.andrew.cmu.edu`).

**Set up**

1. Download the (380MB) database file with the patent citation graph from `https://www.cs.cmu.edu/~christos/courses/826-resources/DATA-SETS-HOMEWORKS/patents/patents.db`
2. At the unix/linux prompt, open the database with the following command:

   ```
   sqlite3 patents.db
   ```

   which should bring you the `sqlite>` prompt.

**Optional set-up steps**

1. **Sanity checks**:
   (a) the command

   ```
   sqlite> .schema Patents
   ```

   should give:

   ```
   CREATE TABLE Patents( "CITING" TEXT, "CITED" TEXT );
   ```

   (b) Check the count of rows - the command:

   ```
   select count(*) from Patents;
   ```

   should give

   ```
   16522438
   ```

   (= total number of rows)
2. **Fun fact**: At `http://patft.uspto.gov/netahtml/PTO/srchnum.htm`, you can look up for the full info about each patent (title, year, inventors, e.t.c.). This could help you double-check the correctness of your responses.

**Data description:** The `patents.db` database has one table `Patents`, listing which patent cites what patent. For example the following row in the table means that patent number `5856190` is citing patent number `4216617`.

```
CITING      CITED
----------  ----------
5856190     4216617
```

**Queries, and what to hand in:** For all the queries below, hand in hard copy of

- both the SQL **code** of your answer,
- as well as the **output** of your code.

Hint: Use `.headers on` and `.mode column` for easier debugging.

(a) [**10 points**] **Self citation:** Check if there are any self citations. Specifically, report all the patent ids (`CITING`) which cite themselves.

Hint: Eliminate duplicates (if any), with `distinct`.

**Solution:** Code:

```
select distinct CITING from Patents where
    CITING=CITED order by CITING;
```

*Grading info: 5 pts for correct code*
*Grading info: full points for all correct alternatives (using 'views' is fine).*
*Grading info: no partial credit, if there are serious errors.*

**Solution:** Output:

```
CITING
----------
5489070
```

*Grading info: 5 pts for correct answer*
*Grading info: no partial credit, if there are serious errors.*

(b) **[30 points] Top-5 most-cited patents:** Specifically, give the cited patent id (`CITED`) and number of citations it has received. Order the results by most citations first, and then sort by patent id in ascending order.

Hint: use the keyword: `limit`.

*(FYI - Relationship to data mining: Grouping, sorting, and spotting of 'heavy hitters' are vital, for several data mining tasks like information summarization and anomaly detection. )*

**Solution:** Code:

```
select CITED, count(CITING) from Patents
    group by CITED order by count(CITING) desc, CITED limit 5;
```

*Grading info: 20 pts for correct code*
*Grading info: full points for all correct alternatives (using 'views' is fine).*
*Grading info: -1 for each small error (wrong ordering, etc)*
*Grading info: no partial credit, if there are serious errors.*

**Solution:** Output:

```
CITED       count(CITING)
----------  -------------
4723129     779
4463359     716
4740796     678
4345262     658
4558333     654
```

*Grading info: 10 pts for correct answer*
*Grading info: -1 if the ordering is wrong*
*Grading info: no penalty if there are no column headers*
*Grading info: -1 if other small errors*
*Grading info: 1pt per correct tuple, if there are serious errors*

(c) **[50 points] Similar patents**: Given a patent id $X$ (= `5795784`), let's call as *targets* the patents that $X$ cites. Find the patents that cite as many of the '*targets*' as possible. More specifically, for each patent, count the number of '*targets*' it cites, and report the top-5 patents with the highest overlap.

As before, please order the results by the count of '*targets*' (descending), and then by patent id (ascending).

Hint1: if the query is slow, create an appropriate index.

Hint2: Remember to exclude patent $X$ from the response.

*(FYI - Relationship to data mining: Such queries are useful in finding similar items, like similar genes/proteins in bioinformatics, near-duplicate tweets (possibly indicating plagiarism/fraud). Also, they are useful in link prediction and product recommendation, like, say Amazon: '*many people who bought product-Z, also bought product-W*'.)*

**Solution:** Code:

```
select B.CITING, count(*)
from Patents A, Patents B
where A.CITING=5795784 and
    B.CITED = A.CITED and
    B.CITING <> 5795784
group by B.CITING
order by count(*) desc, B.CITING
limit 5;
```

*Grading info: 30 points for code - all correct alternatives are acceptable*
*Grading info: -1 for each small error (eg., wrong ordering, duplicates, etc)*
*Grading info: -15 for incorrect interpretation of question*
*Grading info: no partial credit, if there are serious errors.*

**Solution:** Output:

```
CITING      count(*)
----------  ----------
5856194     732
5575978     61
5482861     59
```

```
5635364     59
5646049     59
```

*Grading info:* *20 points for correct answer*
*Grading info:* *-1 for each small error (wrong ordering, etc)*
*Grading info:* *-10 for output based on incorrect interpretation*
*Grading info:* *1pt for each correct tuple, if there are serious other problems*