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 (e.g., 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

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-trees</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Linear Hashing</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>SQL</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Question 1: B-trees ............................ [5 points]

Consider B-trees of order \( d=2 \) \((2^d+1 = 5 = \text{maximum fanout})\). One such tree is in Figure 1.

![B-tree diagram]

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

\( NO \ \text{NEED} \) 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,\ldots,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 \( h_0(x) = x \mod 7 \); the hash table had \( B=7 \) buckets \((0,1,2,\ldots,6)\), and the split pointer was \( s=0 \).

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

(a) [2 points] Where is the split pointer \( s \)? Give the bucket number it points to \((\text{integer in the range } 0,\ldots,27)\)

(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 \((\text{version } 3.7.17)\), which is available on the andrew unix machines \( \text{ssh unix.andrew.cmu.edu} \).

Question 3 continues...
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.

<table>
<thead>
<tr>
<th>CITING</th>
<th>CITED</th>
</tr>
</thead>
<tbody>
<tr>
<td>5856190</td>
<td>4216617</td>
</tr>
</tbody>
</table>

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:
```sql
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:
```
<table>
<thead>
<tr>
<th>CITING</th>
</tr>
</thead>
<tbody>
<tr>
<td>5489070</td>
</tr>
</tbody>
</table>
```

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:
```sql
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:
```
<table>
<thead>
<tr>
<th>CITED</th>
<th>count(CITING)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4723129</td>
<td>779</td>
</tr>
<tr>
<td>4463359</td>
<td>716</td>
</tr>
<tr>
<td>4740796</td>
<td>678</td>
</tr>
<tr>
<td>4345262</td>
<td>658</td>
</tr>
<tr>
<td>4558333</td>
<td>654</td>
</tr>
</tbody>
</table>
```

Question 3 continues...
(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).

**Hint 1:** if the query is slow, create an appropriate index.

**Hint 2:** 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:
```sql
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:

<table>
<thead>
<tr>
<th>CITING</th>
<th>count(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5856194</td>
<td>732</td>
</tr>
<tr>
<td>5575978</td>
<td>61</td>
</tr>
<tr>
<td>5482861</td>
<td>59</td>
</tr>
</tbody>
</table>

Question 3 continues...
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

<table>
<thead>
<tr>
<th>5635364</th>
<th>59</th>
</tr>
</thead>
<tbody>
<tr>
<td>5646049</td>
<td>59</td>
</tr>
</tbody>
</table>