Charles Nguyen -- 011606177

CptS 451 -- HW6

Question 1: Indexing

SQL

```
CREATE TABLE Prof(
        ssno PRIMARY KEY,
        pname,
        office,
        age,
        sex,
        specialty,
        dept_did
);

CREATE TABLE Dept(
        did PRIMARY KEY,
        dname,
        budget,
        num_majors,
        chair_ssno
);
```

1. List the names, ages, and offices of professors of a *user-specified sex (male or female)* who have a *user-specified research specialty (e.g., artificial intelligence)*. Assume that the university has a diverse set of faculty members, making it very uncommon for more than a few professors to have the same research specialty.
    - attributes: <sex, specialty>
    - unclustered
    - hash
2. List all the department information for departments with professors in a *user-specified age range*.
    - attributes: age
    - clustered
    - tree
3. List the department id, department name, and chairperson name for departments with a *user-specified number of majors*.
    - attributes: num_majors
    - unclustered
    - hash
4. List the *lowest budget for a department* in the university.
    - attributes: budget
    - unclustered

    - tree

5. List all the information about professors *who are department chairpersons*.

    - attributes: chair_ssno
    - unclustered
    - hash

Question 2: Storage & Indexing

```sql
CREATE TABLE Student (
        sid PRIMARY KEY, -- 40B
        sname,           -- 40B
        major,           -- 40B
        email            -- 40B
);
```

- The sid is a key (i.e., sid values are unique).
- Assume sid values are uniformly distributed between ‘100’ and ‘204,900’.
- All attributes have type char(40) (i.e., each attribute’s size is 40 bytes).
- The relation contains 100,000 records (assume fixed length records).
- Block size is 16KB+8byte (assume each page has additional 8 bytes to store the pointer to next page).
- Assume the time to read/write to/from a page is D; assume the records are compacted and there is no gap between records.
- Assume each record pointer (RID) size is 8 bytes.
- Assume 1KB= 1000 bytes.

a. Assume relation Student is stored in a heap file. What is the cost of

- (i) file scan, cost $= BD$
    - Each page can pack atmost
        - $16KB = 16384B$
    - Each record requires
        - $160B$
    - Each page can pack
        - 102 records
    - Total number of pages
        - 981
    - cost $= 981D$
- (ii) equality search (sid=’25700’), cost $= 0.5BD$
    - cost $= 491D$
- (iii) range search (sid<=’25700’) on Student? cost $= BD$
    - cost $= 981BD$

b. Assume there is a *clustered B+ tree index* on sid using alternative-1 for relation Student. What is the cost of

- (i) file scan, cost $= 1.5BD$
    - cost $= 1472D$

- (ii) equality search (sid='25700'), cost $= Dlog_F 1.5B$
  - where $F = 100$ typically
  - cost $= D \cdot log_{100}(1.5 \cdot 981) \approx 2$
- (iii) range search (sid<='25700') on Student? cost $= Dlog_F 1.5B + B_{matched}$
  - (assume the B+tree has 67% occupancy, i.e., the physical data pages are 1.5 times more than original data file; assume the height of the B+tree is 3.)
  - cost $= D \cdot log_{100}(1.5 \cdot 981) + B_{matched} \approx 2 + matching\_pages$