CSE 132C
Database System Implementation

Arun Kumar

Exercise 2 with Answers

Time tip: Roughly 1min per 1pt
Q1) [4 x 6pts] Consider the following B+ tree index. Each page can hold 4 entries. Draw the index after each of the following operations has finished (each bullet is independent). Follow all conventions for tree indexes given in the lectures.

A. Insert 23*
B. Insert 22*
C. Delete 34*
D. Delete x WHERE x MOD 5 = 0
Exercise

A. After Insert 23*
Exercise

B. After Insert 22*

Overflow
Exercise

C. After Delete 34*
Exercise

D. After Delete x WHERE x MOD 5 = 0
Exercise

Q2) [2 x 8pts] Consider the following B+ tree index. Each page can hold 4 entries. Draw the index after each of the following sequence of operations have finished (each bullet is independent). Follow all conventions for trees again.

A. Insert 7*; Delete 22*
B. Delete 60*; Insert 80*; Insert 83*
Exercise

A. Insert 7*; Delete 22*
Exercise

B. Delete 60*; Insert 80*; Insert 83*
Q3) [4 x 6pts] Consider the following extendible hash index.

Each bucket can hold 4 entries. Draw the index after each of the following sequence of operations have finished (each bullet is independent).
A. Insert 22*
B. Delete 21*
C. Insert 19*; Insert 35*
D. Insert 7*; Insert 7*
Exercise

A. After Insert 22*

- **Directory**
  - Bucket A
    - 2
    - 4* 36* 40*
  - Bucket B
    - 2
    - 21*
  - Bucket C
    - 2
    - 7* 7* 7*
  - Bucket A2
    - 2
    - 22* 62*
Exercise

B. After Delete 21*

Directory

Bucket A

Bucket C
Exercise

C. After Insert 19*; Insert 35*

000 001 010 011 100 101 110 111

Directory

Bucket A

1
4* 36* 40* 62*

Bucket B

2
21*

Bucket C

3
19* 35*

Bucket C2

3
7* 7* 7*
Exercise

D. After Insert 7*; Insert 7*

```
Directory
  00 01 10 11

Bucket A
  1 4* 36* 40* 62*

Bucket B
  2 21*

Bucket C
  2 7* 7* 7* 7*

Overflow
```

```
Exercise

Q4) [6 x 5pts] You need to sort a file with 20 million pages. You are given B buffer pages for this. What is the total number of passes needed for the following EMS regimes for B = 100? What if B = 5000? What if B = 30 million?

A. Standard EMS  
B. Do internal replacement sort  
C. Do double buffering  
D. Do blocked I/O with b = 10  
E. Do both B and C  
F. Do all of B, C, and D

Number of passes =  

\[ 1 + \log_F(N') \]
### Exercise

<table>
<thead>
<tr>
<th>N = 20M</th>
<th>B = 100</th>
<th>5K</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Standard EMS</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>B. Do internal replacement sort</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>C. Do double buffering</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>D. Do blocked I/O with b = 10</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>E. Do both B and C</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>F. Do all of B, C, and D</td>
<td>10</td>
<td>3</td>
</tr>
</tbody>
</table>

When B = 30M, we can just read the whole file into RAM and do regular internal sort; so, just 1 pass.
Exercise

Q5) [5pts] What is the largest file size you can sort with just 2 passes given B buffer pages for EMS? Use any optimization(s) from those discussed in the lectures. Write the closed form expression in terms of B.

To get largest file size $N$, we need the log input to be lowest and log base to be highest. So, we use internal replacement sort and do not use double buffering or blocked I/O.

$$2 = 1 + \log_{B-1}\left(\lfloor N/2B \rfloor\right)$$

$$\implies N = 2B(B - 1)$$
(Extra Credit; Optional) Q6) [6pts] Suppose you are given a B+ tree index of height 2h and order d. The fanout of each node at the top h levels is d; the rest have fanout 2d. Level 0 is the root. What is the total number of nodes in this index? Write the closed-form expression in terms of h and d.

Number of nodes = \[ 1 + d + d^2 + \cdots + d^h \]
\[ + d^h (2d) + d^h (2d)^2 + \cdots + d^h (2d)^h \]
\[ = \frac{d^{h+1} - 1}{d - 1} + 2d^{h+1} \frac{(2d)^h - 1}{2d - 1} \]
(Extra Credit; Optional) Q7) [8pts] Suppose you are given a valid extendible hash index whose GD is k (>=2). What is the minimum possible number of unique bucket pointers in the directory?

Clearly for GD to be k for a valid index, at least one bucket must have LD k. By extension that means it must have a split image, also with with LD k. So, we need to figure out how few unique pointers we can get away with for the remaining $2^k - 2$ slots in the directory.

Recall that if LD < GD, then $2^{(GD-LD)}$ pointers point to that bucket. We now use this observation to replicate pointers to lower LDs.

WLOG, let the bucket(s) with LD k be “000…0” (and “100…0”); note that each string is k bits.

Send all entries with LSB 1 to a unique bucket, i.e., “[any k-1 bits]1”, which means this bucket has LD 1.

Of what remains, send “[any k-2 bits]10” to a unique bucket with LD 2.

Similarly, send “[any k-3 bits]100” to a bucket with LD 3 and so on until “[any 1 bit]100..0” is sent to a unique bucket with LD k-1.

So, min. total number of unique pointers = $2 + 1 \times (k-1) = k + 1$. 