Elijah Andrushenko

04/27/2019

Introduction to Databases

CPTS 451

Homework 6

Q1.

(A)Record size = 4 \* 40 = 160 Bytes

B = number of pages for the student table on the disc

16,000,000 / 16,000 = Byte for total Data = 1000 pages

(i) Cost of file Scan

1.5 \* 1000 = 1500 pages -> **1500D**

(ii) Equality Search

1

(sid=’25200’)

**3D**

3

2

…….

…….

(iii) Range Search (sid <= ‘25700’)

1500 pages sid values from 100 to 204900

My range condition 100 – 25700

X = ((25700 – 100) \* 1500) / (204900-100) = 8 -> 1500/8 = 188 = number of matching pages

Cost = height of tree + number of matching pages = 3 + 188 = **191D – 192D**

(B) Two cases sorted and unsorted

Record size = 1 \* 40 = 40 Bytes

B = 4,000,000 / 16,000 = 250,000

(i) Cost of file Scan

Unsorted = BD = **250D** Sorted = 4BD = 4\*1000\*D = **1000D**

(ii)Equality Search

D(log10((0.15)250)+1) = D(log10(37.5)+1) = 1.57 + 1 = 2.57 -> **2D**

(iii)

31 matching records + 2 from Equality Search

Cost = height of tree + number of matching pages = 2 + 31 = **33**

Q2.

1. –specialty and rank on the prof relation unclustered Hash Index

-Since its equality search is a constant \* D rather than logarithmic \* D like the trees are. We index on Specialty and Rank since those are what the queries will run on.

2. –age on the prof relation unclustered B+ tree

-Since its range search is the fastest and we run the range search on age so the index needs to be made on age

3. –num\_majors on the dept relation unclusterered Hash Index

-We are running a search so this is the best way to go, see #1 for the explanation

4. –chair\_ssno on the dept relation unclustered Hash Index and ssno on the prof relation unclustered Hash Index

-Since we are searching this is the same solution explaination as #1.

5. –did on the dept relation unclustered hash index and rank and dept\_did from the prof relation unclustered B+ tree

-Hashing the did in depts. And then using a unclustered tree on the prof relation works best since did is a primary key for our index so it is unique, since the other isn’t a key the unclustered tree is better for that part.

6. -did and num\_majors on the dept relation unclustered B+ tree

-go to the node with the lowest number of majors and return that, which should be the most efficient time

7. – age on the prof relation unclustered B+ tree and chair\_ssno on the dept relation unclustered Hash Index and ssno on the prof relation unclustered Hash Index

-getting the age for the youngest professors and unclustered B+ trees are the best for that kind of search. Then we implement a way to search the chair ssno with a hash index since it’s a primary key.