**CS521- Fall 2012 – Problem Set #4 Answers**

**Question 1.**

 Consider following B+ tree



A)  Show the resulting tree after inserting the search key 12 into the given tree.



B)  Show the resulting tree after inserting the search key 32 into the given tree.



C)  Show the resulting tree after inserting the search key 32, and then deleting the key 64 from the given tree.



D)  Show the resulting tree after deleting the search key 64, and then inserting the key 32 into the given tree.

 

**Question 2.**

Use extendible hashing with the initial hash function h(x) = x mod 4 and a bucket size of 3 to create an index on a relation with the following search keys:

2, 3, 5, 7, 11, 17, 19, 23, 29, 31

A) Show the final directory and data pages after inserting the given keys.



B) Show the result of deleting 11 from the hash index that you derived in part A.



C) Show the result of deleting 31 from the hash index that you derived in part A.



D) Show the result of inserting 1 into the hash index that you derived in part A.



E) Show the result of inserting 15 into the hash index that you derived in part A.



**Question 3.**

The following query finds those actors who played a title role in a movie that was made since 2003.

SELECT A.name,M.title

FROM Actors A, Casts C, Movies M

WHERE A.aid=C.aid AND C.mid=M.mid AND C.role=M.title AND year > 2003;

**Useful observations:**

Obs1. Number of pages for the result of join of Casts and Movies relation on mid column is 250 because for each row in Casts, there is only one row in Movies which matches mid column

Obs 2. Reduction factor of selection “year>2003” is 2/92 (or 0.022) because we can assume years are uniformly distributed and we know that minimum is 1914 and max is 2005. Therefore the expected proportion of records that match selection is (2005-2003)/(2005-1914+1) = 2/92 ~ 0.022

Obs 3. reduction factor of selection “role=title” is x (we just give it a name)

**A)** Assume there are no indices. Estimate the number of blocks read for each of the three given plans using simple nested loops.

simple nested loop cost = M + M\*N where M and N are number of pages for relations.

A. (50+50\*250) + (100+100\*250) = 37650

Only joins cost I/O here. Selection and projection are done on the fly

B. (50+50\*250) + [250\*(0.022\*x) + 250\*(0.022\*x)\*100] = a number between 12550 and 13000 (0 <= x <=1)

Again only joins cost I/O here. Cost of first join is obvious. Cost of second join depends on number of pages after selection. From observations above the number of pages after selection on join is 250\*0.022\*x

C. 50 + (50\*0.022+ 50\*0.022\*250) + [250\*(0.022\*x) + 250\*(0.022\*x)\*100] = 325 + 5555x

There is a cost of scanning movies table. We need to scan through all pages so cost is 50

Other costs are joins.

**B)** Estimate the number of blocks read for each of the plans using Sort-Merge-Joins (once again assuming no indices).

Cost of Sort-Merge-Join: MlogM + NlogN + M +N (external sort is also valid)

This part is very similar to A except that cost of joins are calculated using above formula

1. (250log250 + 50log50 + 250+50) + (100log100+250log250+100+250) = 5581
2. (250log250 + 50log50 + 250+50) + (100log100+5.5log5.5x + 5.5x+100) = a number between 3339-3358
3. 50 + (250log250 + 1.1log1.1 + 250+1.1) + (100log100+5.5log5.5x + 5.5x+100) = a number between 3057-3076

**C)** Estimate the number of blocks read for each plan if one takes advantage of a hash index built over the mid attribute of the Casts relation.

Index nested loop (INL) is used for joins involving Casts relation. Cost of INL is M + M\*1.2

We assume simple nested is used loop for other loops (Sort-Merge-Join is also ok)

This part is very similar to A except that cost of joins where C are calculated using above formula

1. (250+250\*1.2)+(100+100\*250) = 25650
2. (250+250\*1.2)+(5.5x+5.5x\*100) = between 550-1105
3. 50 + (250+250\*1.2)+(5.5x+5.5x\*100) = between 600-1155

**Question 4.**

Suppose you have a file with 20,000 pages and you have five buffer pages. Answer the following questions for given scenario, assuming that the most general external sorting algorithm is used:

A) How many runs will you produce in the first pass?

ceil(N/B) = ceil(20,000/5) = ceil(4,000) = **4,000 runs**

B) How many passes will it take to sort the file completely?

1+ceil(logB-1(ceil(N/B))) = 1 + ceil(log4(4,000)) = **7 passes**

C) What is the total I/O cost of sorting the file?

2N \* # passes = 2N \* [1 + ceil(logB-1(ceil(N/B)))] = 2\*20,000\*[1+ ceil(log4(4,000))] = 2\*20,000\*7 = **280,000 I/Os**

D) How many buffer pages would be needed to sort the file completely in just two passes?

1+ceil(logB-1(ceil(N/B))) <= 2

ceil(logB-1(ceil(N/B))) <= 1

⇒ **B >= 142** ⇒ **at least 142 buffer pages are required to sort the file in 2 passes**