Questions

1. B+ Tree(logF), Linear Search, Binary Search
2. Query , range query based on primary index
3. Secondary index
4. Unindexed
5. Join
    a. Hashed join
    b. Join in main memory
    c. Natural join, inner join, outer join
6. How about joining three tables?
7. Comparison with KD-Tree in search time

Out of syllabus: Sorting, nested loop join, merge sort join

Always ceil number of blocks, tuples.

Sc = n/V(a,r) = average number of tuples for age 21
sc/fr = number of blocks for age 21
nr/fr = total number of blocks

B+ Tree, Normal Search Query, Primary Index

Number of blocks need to be access->

Linear Search = Br
Binary search = log2(Br) + Sc/fr - 1
B+ Tree = logF(V(a,r)) + sc/fr , F = Fanout value

B+ Tree, Normal Search Query, Secondary Index

Primary index has clustering property.
Abhijit,rahim,rahim,rahim,abhijit
9,10,10,10,21
Secondary indexes don't have clustering properties.

Linear Search = Br
Binary search = log2(Br) + Sc - 1 [not sure]

B+ Tree = logF(V(a,r)) + sc(A,r)

Normal Search, Unindexed

Search = br
Search = log2(br) if search key = primary key

Range Query

```
Select * from user where age > 21;
```

Primary index & primary key :
B+ Tree = logF(V(age,user)) + 1

Primary Index & non primary key:
B+ Tree = logF(V(age, user)) + br/2;

Secondary Index:
B+ Tree = logF(V(age, user)) + V(a,r)/2 + nr/2

Unindexed:
Search = nr/3

Statistical Information:
nr*(v-min)/(max-min)

Example:
Select * from student where student_name = labonya;
Student_name is the primary index, not primary key.
logF(V(a,r)) + sc/fr

Join

T(A) = number of rows in Table A

Join Tuple Size = T(A)*T(B)/max(V(id,A), V(id,B))

10^2 +

Question Solve

We want to find the cost of JOIN

Fr = 10

T(webpage) = 10^2
V(webpage, url) = 10^2 [Filter korar por paisi]
B(Webpage) = 10^2/10 = 10

T(occurs) = 10^12
V(occurs, url) = 10^9
B(Webpage) = 10^12/10 = 10^11

Question 1:
Webpage.url = Primary Index
Occurs.url = Secondary Index

Filtered tables will always be **unclustered [Webpage]**

Formula = T(webpage)*T(occurs)/max(V(webpage,url), V(occurs, url))
= 10^2*10^12 / max(10^2, 10^9)

Question 2:
Webpage.url = Secondary Index
Occurs.url = **Primary** Index

Filtered tables will always be **unclustered [Webpage]**

Formula = T(webpage)*B(occurs)/max(V(webpage,url), V(occurs, url))
= 10^2*10^11 / max(10^2, 10^9)

Tuple Reading Time