Sandeep Neupane

1.

Solution :

1. The condition is : if v > max(A,r)
2. C = nr\* (max(A,r)-v)/(max(A,r)-min(A,r))

2.

Solution:

1. If R ^ S = null, This means that the join is not meaningful and number of tuple in this join is 0
2. If R ^ S is a key for R then , then a tuple of *s* will join with at most one tuple from *r ,*

*So max number of tuples = number of tuples in s = 20000*

C*. if R ^ S is a foreign key in S referencing R,*

*Number of tuples in this join = number of tuples in s = 20000*

*D. if R ^ S = {A} is not a key for R or S*

*lIf we assume that every tuple t in R produces tuples in R join S, the number of tuples in R \* S is estimated to be: (nr\*ns)/(V(A,s)) =*

*= 10000\*20000/5000*

*= 40000*

*If the reverse is true, the estimate obtained will be: (nr \* ns)/(V(A,r))*

*= 10000\*20000/10000  
 = 20000*

3*.Suppose that a B+-tree index on (dept\_name, building) is available on relation department which has dept\_name, building and budget as its attributes. What would be the best way to handle the following selection?*

*σ (building < “Watson”) ^ (budget < 55000) ^ (dept\_name = “Music”) (department)*

*Solution :*

*Since we have index on dept\_name ,*

*We would need to evaluate the two conditions ( dept\_name = “music” & building <= “watson” ) first . Then we can evaluate the attribute budget < 55000.*

*4.*

*Solution a.*

*I would sort r based on the join attribute and then take the top “k” tuples.*

*Now I will join these top k tuples with s.*

*Solution b.*

*For this, I would sort r based on the join attribute and then join the first tuple r followed by join of second tuple s. I would continue the above step until, top k tuples are found as required.*