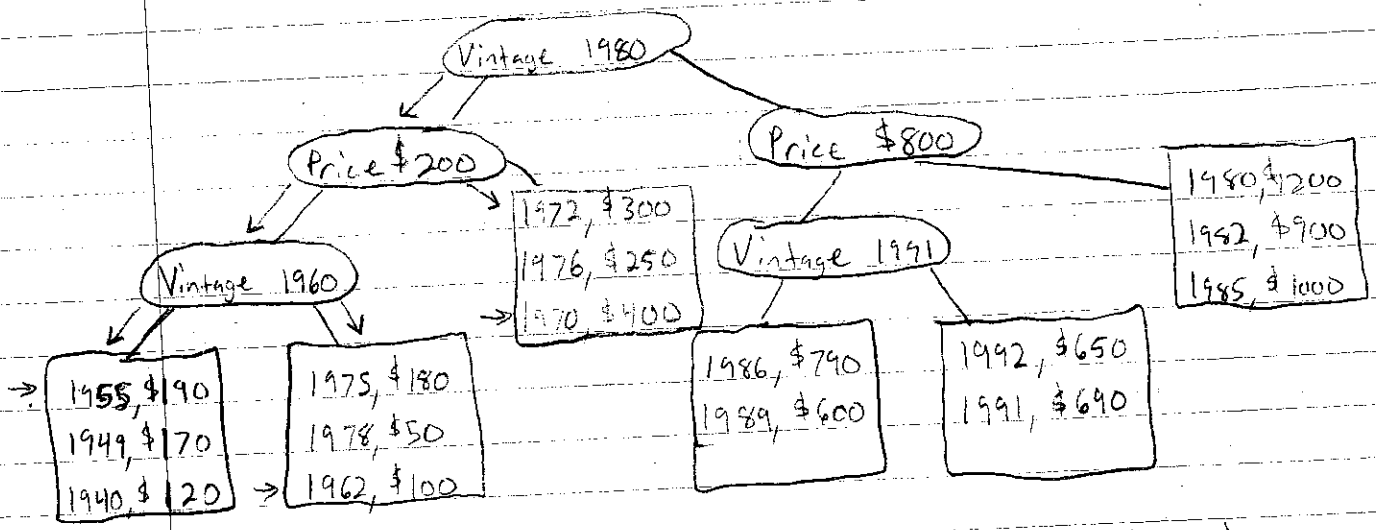
Problem # 1

Start at root node.

1. Compare node attribute & value with corresponding attribute of range query.
   - case: Root does not split query range.
     Traverse appropriate branch
   - case: Root splits query range
     Traverse both sides of branch.

2. Repeat until leaf node is reached
   - case: Leaf node is within both query ranges
     Print & return values.
   - case: Leaf node is not within both query ranges
     Ignore leaf node

ex: Wines (Vintage, Price) ★block size of 3



Query: (Vintage 1950 - 1970, Price $1 — $400)

Ans: 1955, $190
1962, $100
1970, $400

Practice Midterm

#2 Bitmap Indexes

There are 9 records
3 unique values for A
5 unique values for B

Col A Bitmap Index:

1 111 060 000
2 000 111 000
3 000 000 111

Col B Bitmap Index:

a 100010000
b 010001000
c 001000110
d 000100000
e 000000001

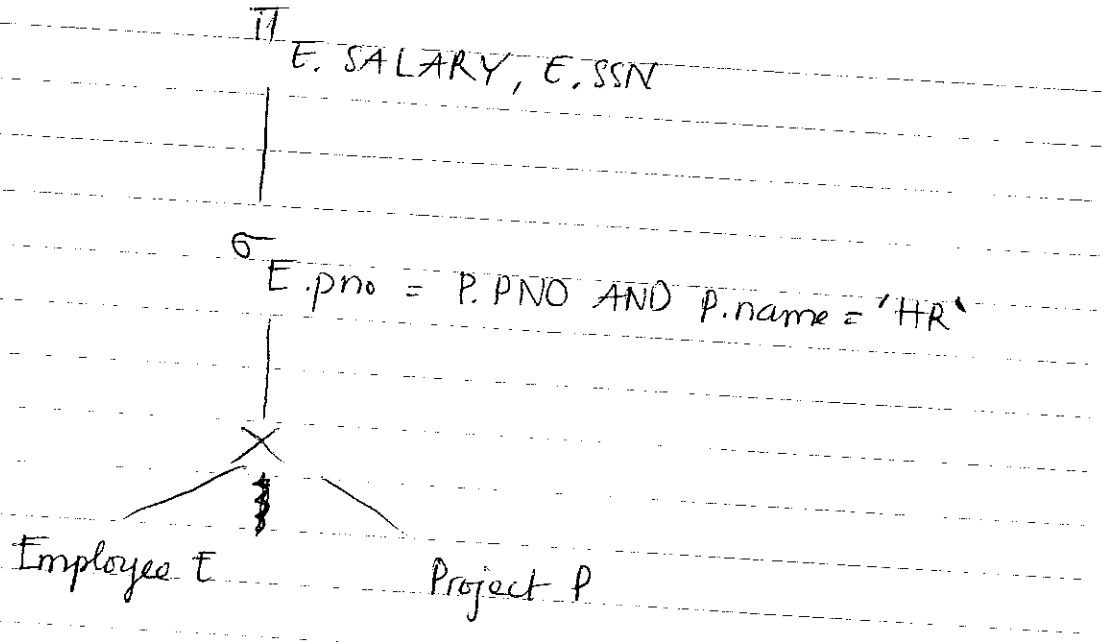
3. bag Union

$R \cup S = \{A, A, A, B, B, C, C, C\}$.

Product

$R \times S = \{\langle A,C\rangle, \langle A,A\rangle, \langle A,A\rangle, \langle A,B\rangle,$
$\langle B,C\rangle, \langle B,A\rangle, \langle B,A\rangle, \langle B,B\rangle,$
$\langle C,C\rangle, \langle C,A\rangle, \langle C,A\rangle, \langle C,B\rangle,$
$\langle C,C\rangle, \langle C,A\rangle, \langle C,A\rangle, \langle C,B\rangle\}$.

#4



Shruthi Gaonkar
Vasuki Kallur
Rita Ho

5.) An iterator is an object which supports the following operation
GetNext() - get next result (usually a tuple) from results
open() - position iterator are start of results
close() - stop using iterator
Iterators are used for table & index scans as well as pipelining in query evaluation

$B(\delta(R)) \leq M-1$ & $B(\delta(R)) \leq M$

1.) Read R into main memory
2.) You have to read each tuple into the main memory. The first tuple you have write to the output. I keep a list of all the tuples read in some kind of a binary structure. So each time you read a tuple see if it is there in the Binary tree structure. If it is already there, then I do not write it to the output. If it is not there, write it to the output.
It can be used for queries like $\delta(R)$

One memory buffer should be used to ~~read~~ read in each block of R. The remaining M-1 buffer can be used to store the copy of the tuples seen.
The context it can be used in to convert bag into a set using the DISTINCT.

10.) ~~$\Pi_L(R \bowtie S) = \Pi_L(\Pi_M(R) \bowtie \Pi_N(S))$~~

$\Pi_L(R \bowtie_C S) = \Pi_L(\Pi_M(R) \bowtie_C \Pi_N(S))$

~~$\Pi_L(R \times S) = \Pi_L(\Pi_M(R) \times \Pi_N(S))$~~

~~$\Pi_L(R \cup_B S) = \Pi_L(R) \cup_B \Pi_L(S)$~~

~~$\Pi_L(\sigma_C(R)) = \Pi_L(\sigma_C(\Pi_M(R)))$~~

6. Describe w/ an example how the two pass hash-join algorithm works?

- Use only the join attri as hash key.
- Hash R and S to hash table.
- Do one pass join of bucket 1's, 2's etc

ex. R⋈S on Name

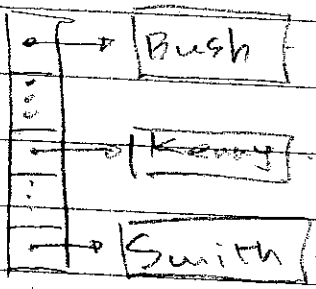
R:

Name	Dept
Smith	Bio
Bush	Math
Kenny	Law

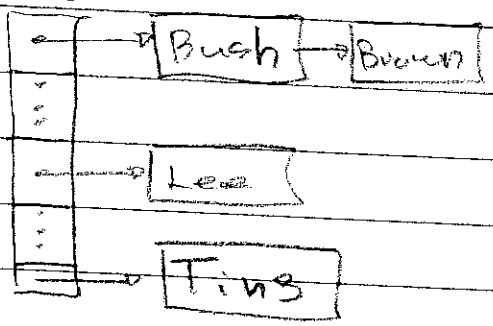
S:

Name	Score
Brown	90
Lee	85
Ting	99
Bush	50

R:



S:



- When do one pass join:

Put R's 1st bucket into Mem

Put S's 1st bucket into Mem

- concatenate each of R's 1st bucket w/ the appropriate tuples from S's 1st bucket.

R⋈S:

Name	Dept	Score
Bush	Math	50

- So, will find

// For each pointer position in R's bucket, go through the entire corresponding S's bucket chain by advancing the pointer.

- will find the pair that we are looking for.

7) Using formula $\frac{R(A,B) \times S(B,C)}{MAX(V(R,B), V(S,B))}$

we pick: $V(S,B) = 30$ for:

$$\frac{1000 \times 2000}{30} = \sim 67{,}000 \text{ tuples}$$

8.) What is a left-deep join tree?
What is branch and bound plan enumeration?

* A left-deep join tree is a binary tree with all right children as leaves. There can be interior nodes, with operators other than a join. It interacts well with common join algorithms, such as nested-loop joins and one-pass joins, because it is easier to implement pipelining compared to pipelining in balanced trees.

* Use heuristics to find a good physical plan for the entire logical query plan. Let the cost this plan be C. Now branch over possible subplans for this logic plan.
If you obtain a new plan for the complete query with cost less than C, then we replace C by the cost of the better plan in the subsequent exploration of the span of physical query plans.
If the cost of a subplan is ever greater than C, then eliminate any subquery plans that contain this one, since no complete query plan for that subquery can be better than C.

9 A sort join is good choice when either:

1) One or both arguments are already sorted on their join attribute
2) There are 2 or more joins on the same attribute such as
   $(R(a,b) \bowtie S(a,c)) \bowtie T(a,d)$
   where sorting R and S on a will cause the result of $R \bowtie S$ to be sorted on a and used directly in a second sort-join

2.

A: Value	Vector
1	111 000 000
2	000 111 000
3	000 000 111

B: Value	Vector
a	100 010 000
b	010 001 000
c	001 000 110
d	000 100 000
e	000 000 001

Problem 7)

$$T(R \bowtie S) = \frac{T(R)\,T(S)}{\max(V(R,B), V(S,B))} = \frac{1000 \cdot 2000}{30} = 66667 \text{ tuples}$$

Problem 10) $\pi_L(R \bowtie S) = \pi_L(\pi_M(R) \bowtie \pi_N(S))$

Where M is the list of attr. we proj. out from R and N is the list of attr. we get out of S and L is the list of attr. we want from the joined result.