Database Systems

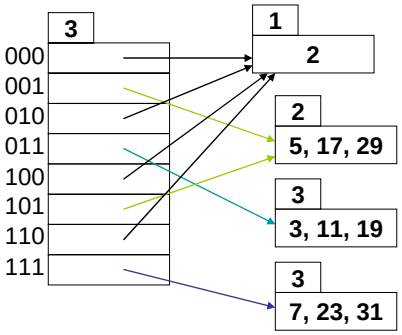
Instructor: Hao-Hua Chu
Fall Semester, 2005

Assignment 5: Hash-based Indexing & Query Evaluation Overview

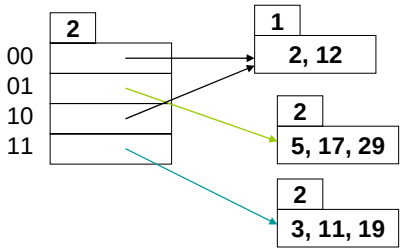
Answers

1.

a. (10%)



b. (10%)



2. (10%) Hash indices enable us to perform point lookup (eg. $\sigma_{A=r}$(relation)) operations very fast, but for range searches the B+-tree index would be much more efficient.
(10%) If there is a range query to be evaluated, and only a hash index is available, the better strategy might be to perform a file scan rather than using that index.

3. a. (10%) Use the index to locate the first tuple whose branch-city field has value “Brooklyn”. From this tuple, follow the pointer chains till the end, retrieving all the tuples.

b. (10%) For this query, the index serves no purpose. We can scan the file sequentially and select all tuples whose branch-city field is anything other than “Brooklyn”.

c. (10%) This query is equivalent to the query

$$\sigma_{(\text{branch-city} >= \text{“Brooklyn”} \text{ AND assets} >= 5000)}(branch)$$

Using the branch-city index, we can retrieve all tuples with branch-city value greater than or equal to “Brooklyn” by following the pointer chains from the first “Brooklyn” tuple. We also apply the additional criteria of assets >= 5000 on every tuple.

4. *r1* needs 800 pages, and *r2* needs 1500 pages

   a. Index Nested Loop (15%)
   Since *r1* is smaller, we use it as the build relation and *r2* as the probe relation.
   The total cost
   = Cost of Scanning *r1* + #*r1* tuples*(Cost of retrieving index page + Cost of retrieving data page)
   = 800 + 20000*(1 + 1)
   = 40800 page I/Os.

   b. Sort-Merge (15%)
   The total cost
   = Cost of sorting *r1* & *r2* + Cost of merging
   = 2*800*(*ceiling*($\log_{100}$ *ceiling*(800/101)) + 1) + 2*1500*(*ceiling*($\log_{100}$ *ceiling*(1500/101)) + 1)
   + (800 + 1500)
   = 11500 page I/Os.