Question 1. [14 MARKS]

Short Answer Question

Part (a) [2 MARKS] What are some of the reasons to use DBMS?

Data independence and efficient access Reduced application development time Data integrity and security Uniform data administration Concurrent access, recovery from crashes.

Part (b) [2 MARKS] what are the main layers of DBMS?

Query Optimization and Execution Relational Operators Files and Access Methods Buffer Management Disk Space Management

Part (c) [2 MARKS] What would be the cost of sorting N pages using B buffers with external sorting?

Please check week 2 slides

Part (d) [2 MARKS] What are statistics and catalogs and how can they help?

week3 slide 6

Part (e) [2 MARKS] Briefly explain the Hash-Join algorithm

week4 slide 8

Part (f) [4 MARKS] Explain 2 algorithms to project distinct values.

week4 slides 15 and 16

Question 2. [12 MARKS]

Cost Calculation

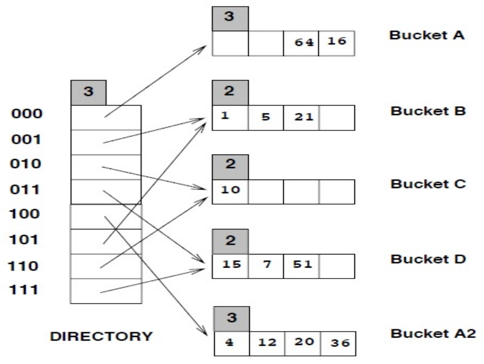
Fill in the blanks in the following table

- B: The number of data pages
- R: Number of records per page
- D: (Average) time to read or write disk page
- Hash: No overflow buckets, 80% page occupancy
- Tree: 67% occupancy

	Scan	**Equality**	**Range**	**Delete**
Heap				
clustered Tree index				
clustered Hash index				

Question 3. [12 marks]

Consider the following Extendible Hashing index and answer following questions



Part (a) [2 marks] What can you say about the last entry that was inserted into the index?why?

It could be anyone of the data entries in the index. We can always nd a sequence of insertions and deletions with a particular key value, among the key values shown in the index as the last insertion. For example, consider the data entry 16 and the following sequence: 1 5 21 10 15 7 51 4 12 36 64 8 24 56 16 56 D 24D 8D The last insertion is the data entry 16 and it also causes a split. But the sequence of deletions following this insertion cause a merge leading to the index structure shown in

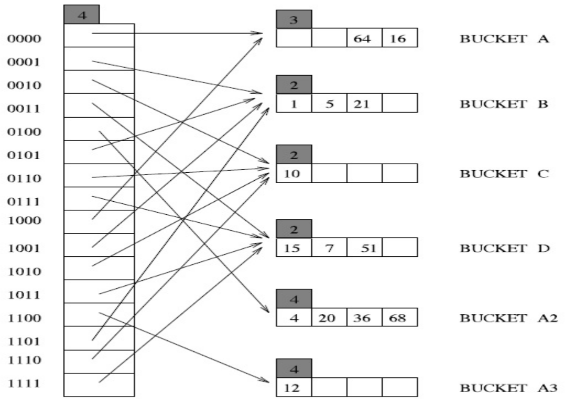
Part (b) [2 marks] What can you say about the last entry that was inserted into the index if you know that there have been no deletions from this index so far?

The last insertion could not have caused a split because the total number of data entries in the buckets A and A2 is 6. If the last entry caused a split the total would have been 5.

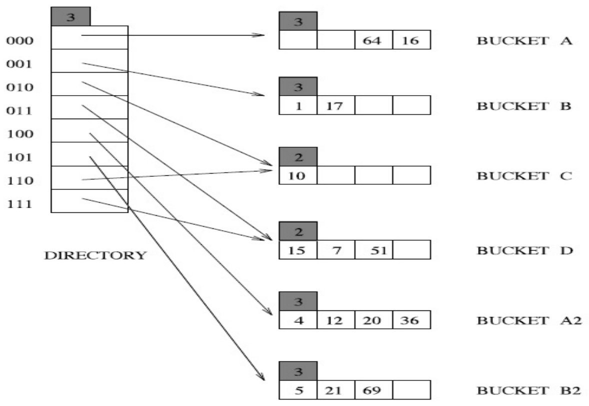
Part (c) [2 marks] Suppose you are told that there have been no deletions from this index so far. What can you say about the last entry whose insertion into the index caused a split?

The last insertion which caused a split cannot be in bucket C. Buckets B and C or C and D could have made a possible bucket-split image combination but the total number of data entries in these combinations is 4 and the absence of deletions demands a sum of atleast 5 data entries for such combinations. Buckets B and D can form a possible bucket-split image combination because they have a total of 6 data entries between themselves. So doA and A2. But for the B and D to be split images the starting global depth should have been 1. If the starting global depth is 2, then the last insertion causing a split would be in A or A2.

Part (d) [2 marks] Show the index after inserting an entry with hash value 68.



Part (e) [4 marks] Show the original index after inserting entries with hash values 17 and 69.



Question 4. [12 marks]

Join Algorithms

Consider the join ($R \bowtie S$) using R.a and S.b columns, given the following information about the relations to be joined. The cost metric is the number of page I/Os unless otherwise noted, and the cost of writing out the result should be uniformly ignored.

- Relation R contains 10,000 tuples and has 10 tuples per page.
- Relation S contains 2,000 tuples and also has 10 tuples per page.
- Attribute b of relation S is the primary key for S.
- Both relations are stored as simple heap files.
- Neither relation has any indexes built on it.
- 52 buffer pages are available.

Part (a) [2 marks] What is the cost of joining R and S using a page-oriented simple nested loops join? What is the minimum number of buffer pages required for this cost to remain unchanged?

$TotalCost = M + (M * N) = 2000010000$

Part (b) [2 marks] What is the cost of joining R and S using a block nested loops join? What is the minimum number of buffer pages required for this cost to remain unchanged?

$TotalCost = N + M * (N/(B2)) = 4200$

B)If the number of buffer pages is less than 52, the number of scans of the inner would be more than 4 since = (200/49) is 5. The minimum number of buffer pages for this cost is therefore 52.

Part (c) [4 marks] What is the cost of joining R and S using a sort-merge join? What is the minimum number of buffer pages required for this cost to remain unchanged?

$TotalCost = 3 * (M + N) = 3,600$

The minimum number of buffer pages required is 25. With 25 buffer pages, the initial sorting pass will split R into 20 runs of size 50 and split S into 4 runs of size 50 (approximately). These 24 runs can then be merged in one pass, with one page left over to be used as an output buffer. With fewer than 25 buffer pages the number of runs produced by the first pass over both relations would exceed the number of available pages, making a one-pass merge impossible.

Part (d) [2 marks] What is the cost of joining R and S using a hash join? What is the minimum number of buffer pages required for this cost to remain unchanged?

$TotalCost = 3 * (M + N) = 3600$

Part (e) [2 marks] What would be the lowest possible I/O cost for joining R and S using any join algorithm, and how much buffer space would be needed to achieve this cost? Explain briefly.

$TotalCost = M + N = 1200$

The minimum number of buffer pages for this cost is $N + 1 + 1 = 202$