

CS3402 Database Systems

Homework 2 Sample Solution

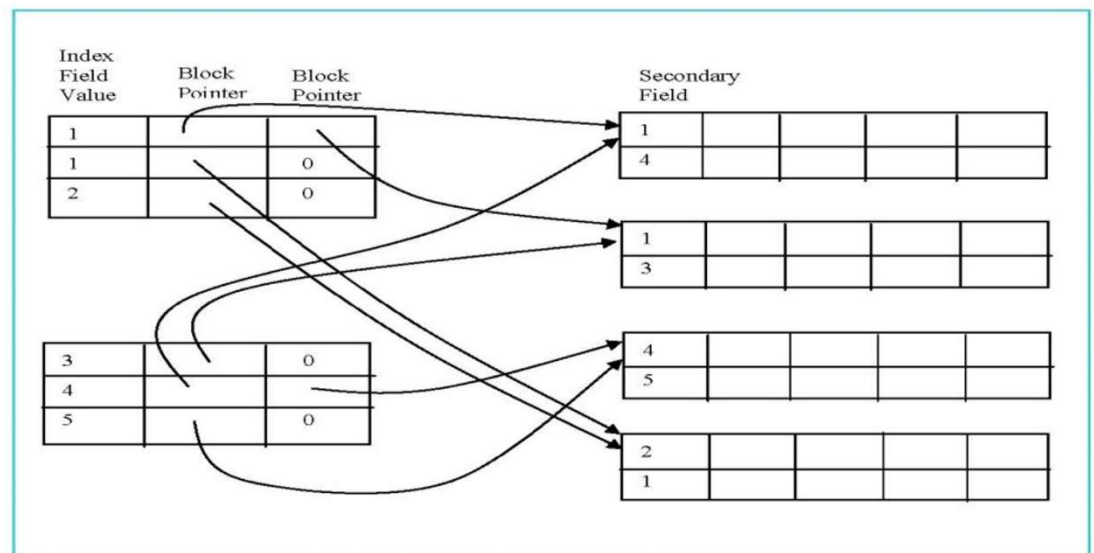
Question A:

- a) $R = 40+8+4+65+8+8+1+7+6+2+1 = 150$ bytes. (5 marks)
- b) The maximum number of accesses is maximum depth of B-tree. For the worst case of the B-tree:
- the root has only 2 pointers, implying $r/2$ records in each subtree;
 - all other nodes are half full (ie, $2 = \text{ceiling}(4/2)$ is the number of pointers for each node).

So, if j is the number of levels, we have: $r/2 = (2)^{j-1}$
 $j-1 = \log_2 (r/2)$
 $j = 1 + \log_2 (r/2)$
 since $r = 100,000$, we have: $j = 1 + \log_2 (50000)$

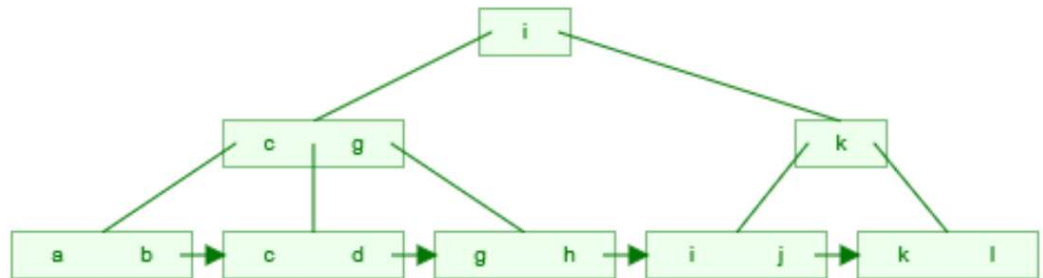
Thus, the number of accesses is $t = \text{ceiling}(j)$
 $= \text{ceiling}(1 + \log_2 (50000))$. (10 marks)

- c) Altogether **10 marks**: each of the pointer gets 1 mark; each "0" get half mark

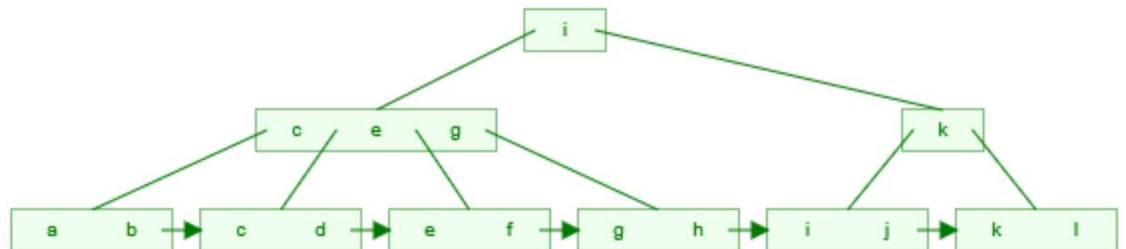


Question B:

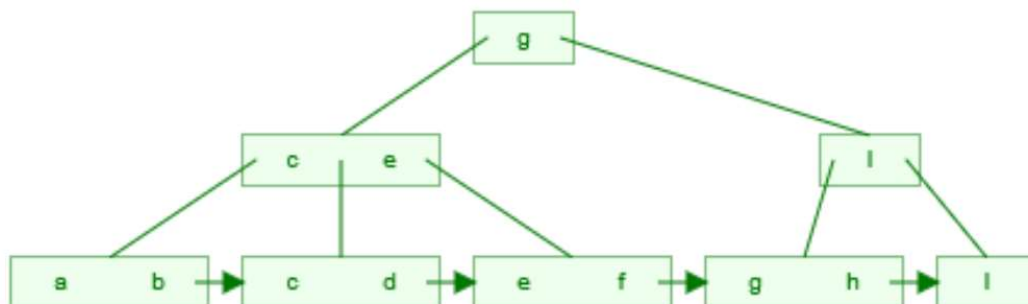
(1) [10 marks]



(2) [8 marks]



(3) [8marks]



Question C:

- 1) (a) Using nested (inner-outer) loop strategy, for each tuple in R1, we must perform an access for each tuple in R2. This involves $20,000 \times 45,000 = 900,000,000$ accesses of tuples in R2. So when we include the 20,000 accesses to read R1, a total of $900,000,000 + 20,000 = 900,020,000$ accesses are required. [8marks]
- (b) If we consider buffering the whole block in memory when reading each block, a total of $900,000,000/30 + 20,000/25 = 30,000,800$ accesses are required. [8 marks]

//either answer gets 8 marks

- 2) Using sort-merge join strategy, if
- (a) the relations are already sorted by the join attributes. In this case we read each block of R1 and R2 only once, therefore $20,000/25 + 45,000/30 = 2,300$ accesses are required; [8 marks]
- (b) the relations are not yet sorted, then we'll need to sort the two tables first, so it requires
- $(800 * \ln 800 + 1500 * \ln 1500) + (800 + 1500)$.

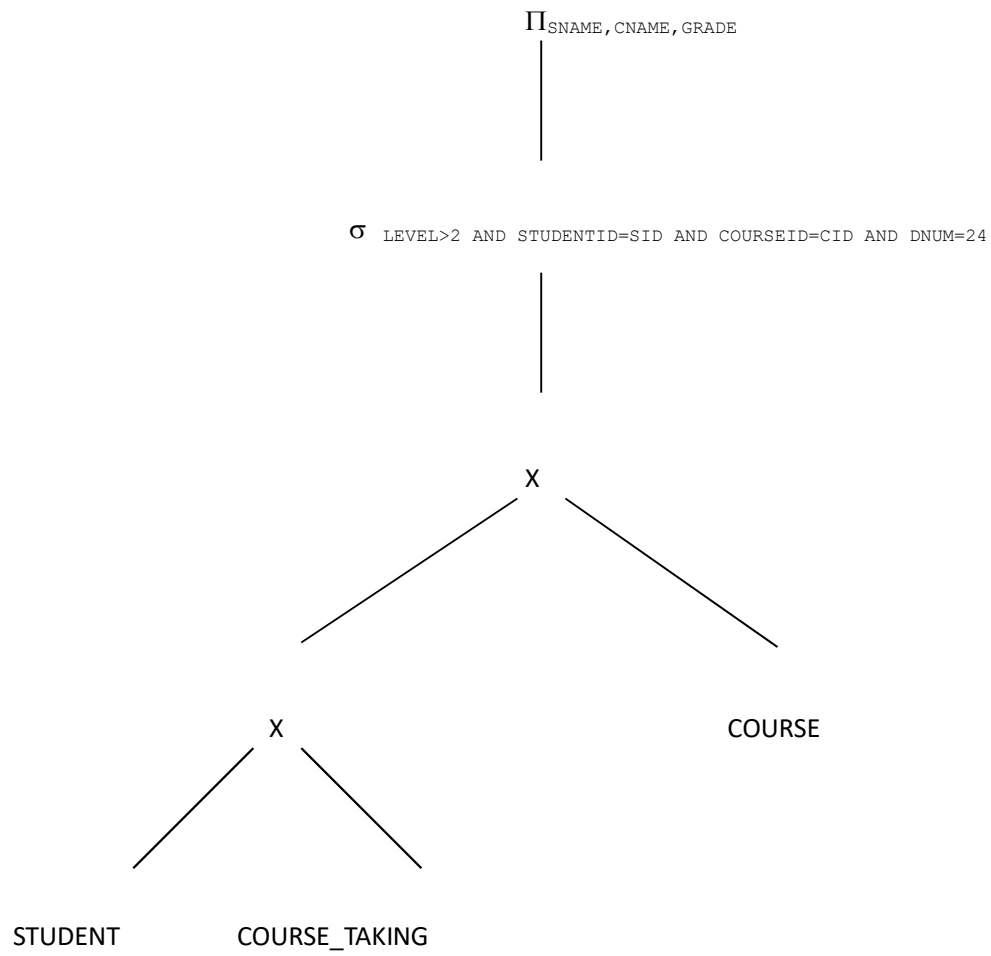
//the 1st parenthesis is for the two tables to sort, and the 2nd parenthesis is for the actual join. [8marks]

//either answer gets 8 marks

- 2) Using hash-join strategy, first a single pass through the file with fewer records hashes its records into the hash buckets. In this stage $20,000/25$ block accesses are required. In the simplest case, we assume that the smaller file can fit entirely in main memory after it is partitioned. In the second phase, a single pass through the other file then hashes each of its records using the same hash function to probe the appropriate bucket, and that record is combined with all matching records in that bucket. In this stage, $45,000/30$ accesses are required. Therefore, totally $20,000/25 + 45,000/30 = 2,300$ blocks access is required. [8 marks]

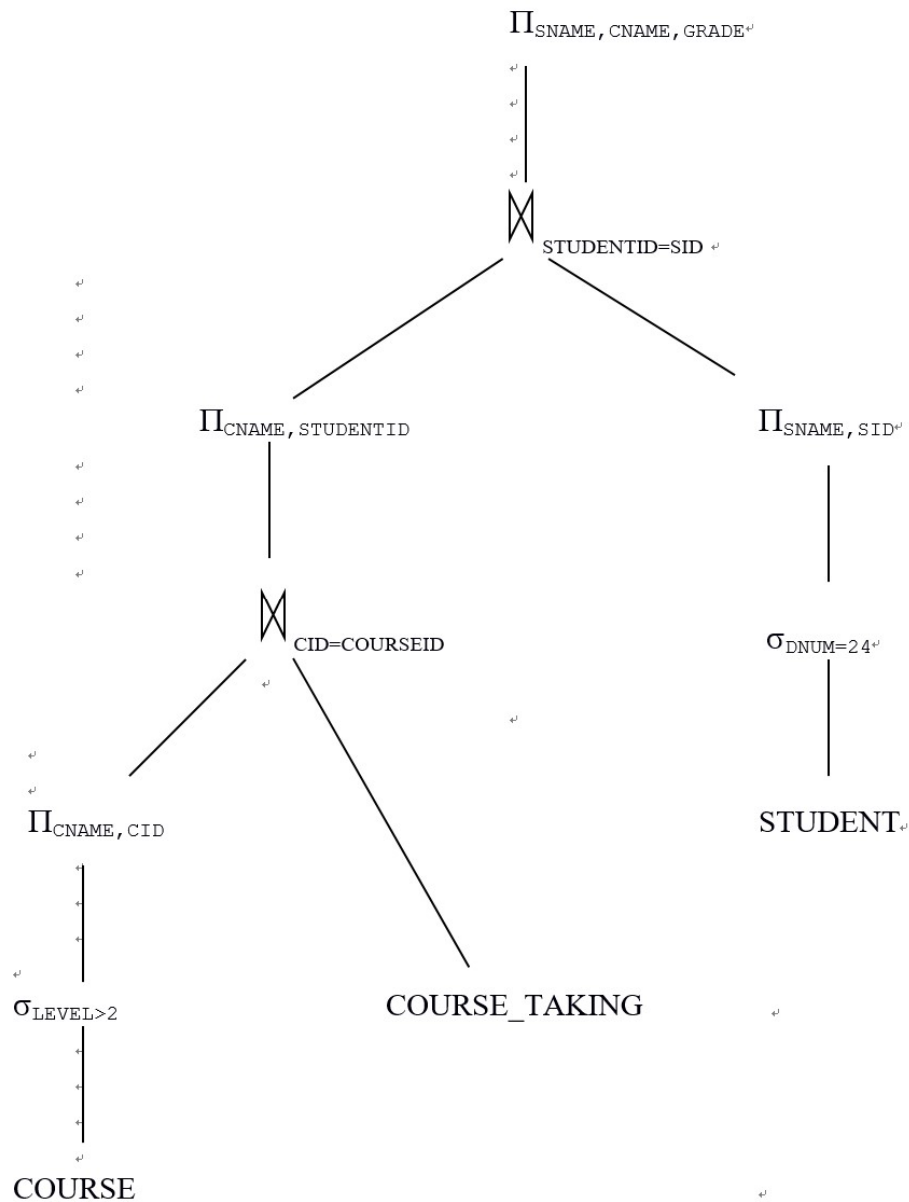
Question D :

(1) [10 marks]



(2) [15 marks]

Assumptions: There are less courses with level being larger than 2 in comparison with the number of students in department 24.



Marking scheme: *Altogether 15 marks. One optimization step missing will get 3 point deducted, and each typo or missing symbol will also cause 1-2 points to be deducted.*