1. (6 points)

Consider a disk with the following characteristics:
The disk rotates at 3000 rpm.
The disk has 7000 tracks per surface. The tracks are numbered from 1 to 7000.
The disk has 8 surfaces.
Each track is divided into 16 sectors, and a sector holds 512 bytes.
Blocks consist of 4 sectors. 20% of the circumference of each track is occupied by gaps.
The time it takes the head to move n tracks is 1+0.003n ms.

a) What is the transfer time for one block? 4.75
b) Assume that the head is currently located at track 7000. We want to fulfil request
Track different from 7000. How many tracks does the head have to move

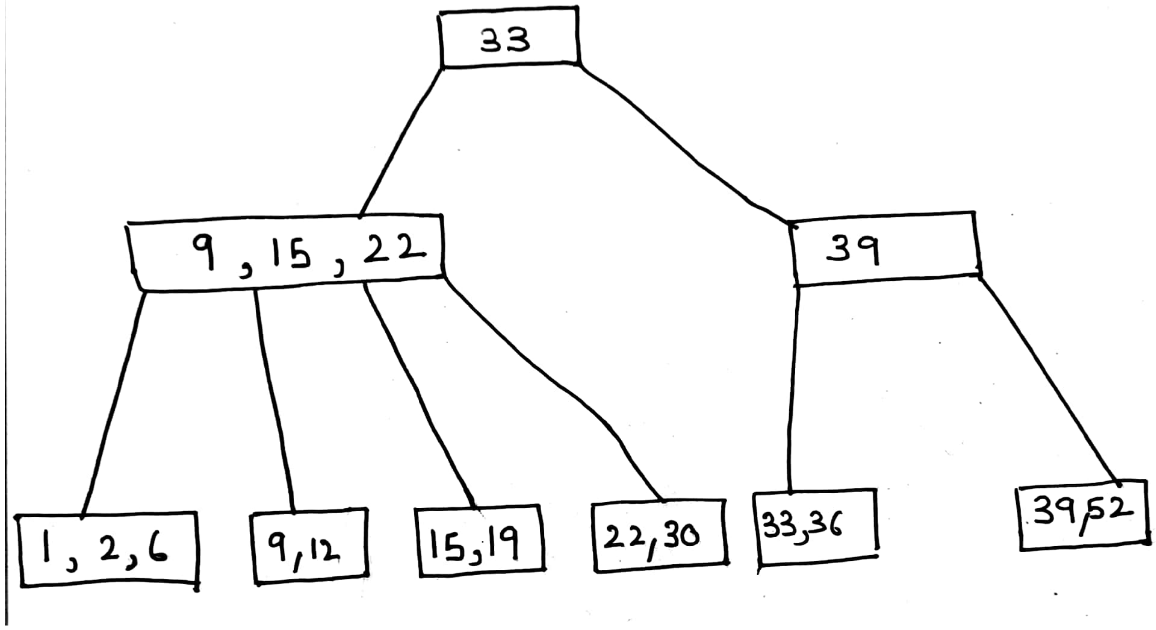
a)

2.

3. (9 points)

Consider B+ tree of order 3

a) Delete from the tree below the key 36.
b) Insert into the tree below the key 5. (Do not use the result of a)!)
c) What is the maximum number of indexed keys we can have in a B+ tree of order 3 with four levels?



4. (13 points)

Suppose blocks consist of 4096 bytes. We want to store the relation R (A, B, C). The size of A is 16 bytes, the size of B is 27 bytes, and the size of C is 35 bytes. Each record consists of 3 data fields and a record header. The record header consists of four fields with 4 bytes for each of them. A block has a block header that consists of eight fields with 6 bytes for each of them. We have T(R)= 100000. For part a), c) and d) fields can start at any byte.

a) what is the minimum number of blocks that we need to store the relation R (A, B, C) if we use spanned storage of records? Assume that the management of spanned records does not require additional space.
b) Repeat a) when we have the additional requirement that fields must start at a byte that is a multiple of 8.
c) We want to create a sparse index for the attribute C on the file created in a). A pointer has a size of 8 bytes. The blocks are described above. The index entries themselves do not have a header. The index entries are stored unspanned. What is the minimum number of blocks that we need for the index file?
d) Consider the query $\pi_{C}(\sigma_{A=17}(R))$. Assume that a projection is not eliminating duplicates. We have V (R, A) =25 and DOM (R, A) = 200. We assume that 17 is an arbitrary domain value. Calculate the minimum number of blocks to store the estimated result relation. The records of the result relation do not have a header. The blocks are as described above. Use unspanned storage of records.

5. (6 Points)

Below are some statistics for four relations W, X, Y and Z.

W (a, b)	X (b, c)	Y (c, d)	Z (c, d)
T(W) = 20	T(X) = 900	T(Y) = 800	T(Z) = 50
V (W, a) = 10	V (X, b) = 10	V (Y, c) = 4	V (Z, c) = 2
V (W, b) = 5	V (X, c) = 30	V (Y, d) = 5	V (Z, d) = 1

We want to perform the natural join for all four relations. What is the join-order selected by the greedy algorithm? What is its cost (measured by the sum of the sizes of the intermediate relations)? Provide the results of all intermediate steps.

6. (6 Points)

We have two relations R (A, B) and S (B, C) with the following statistics:

- T (R) = 120000, B(R) = 6000, V(R, A) =60000, V(R, B) = 10000
- T (S) = 90000, B(S) = 30000, V (S, B) = 90000, V(S, C) = 10000

Assume that the relations are clustered, i.e., are stored contiguously. The attribute B is a key in S and a foreign key in R. For the relation S we have an index on the attribute B. The index has three levels. Ignore the final output I/O cost.

a) We want to perform a natural join of R and S by using an index join algorithm. We assume that the index is completely in main memory. What is the required number of I/O's?
b) We want to perform a natural join of R and S by using an index join algorithm. We assume that the root of index is in the main memory but the blocks from the other levels are not. What is the required number of I/O's?