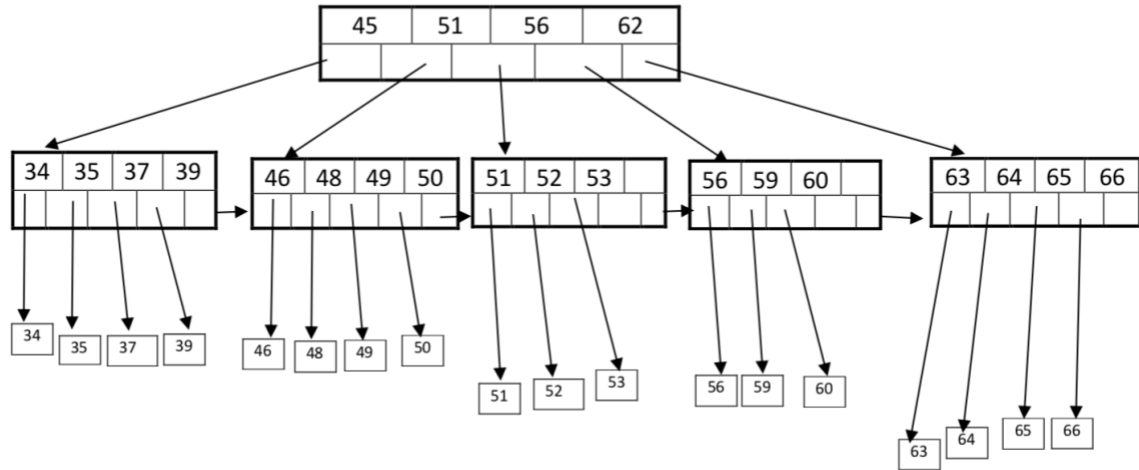
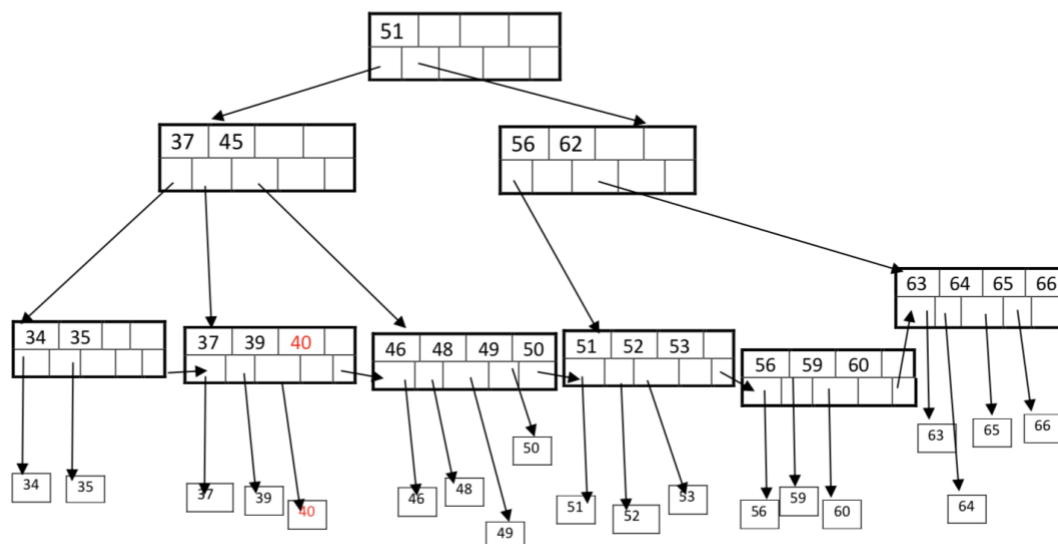


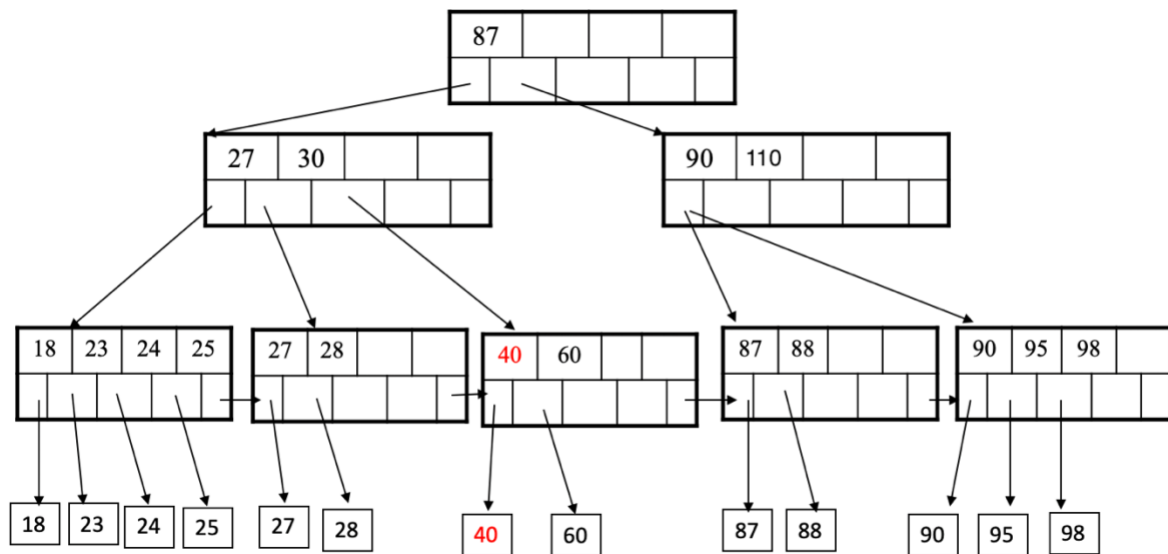
1.
 a)



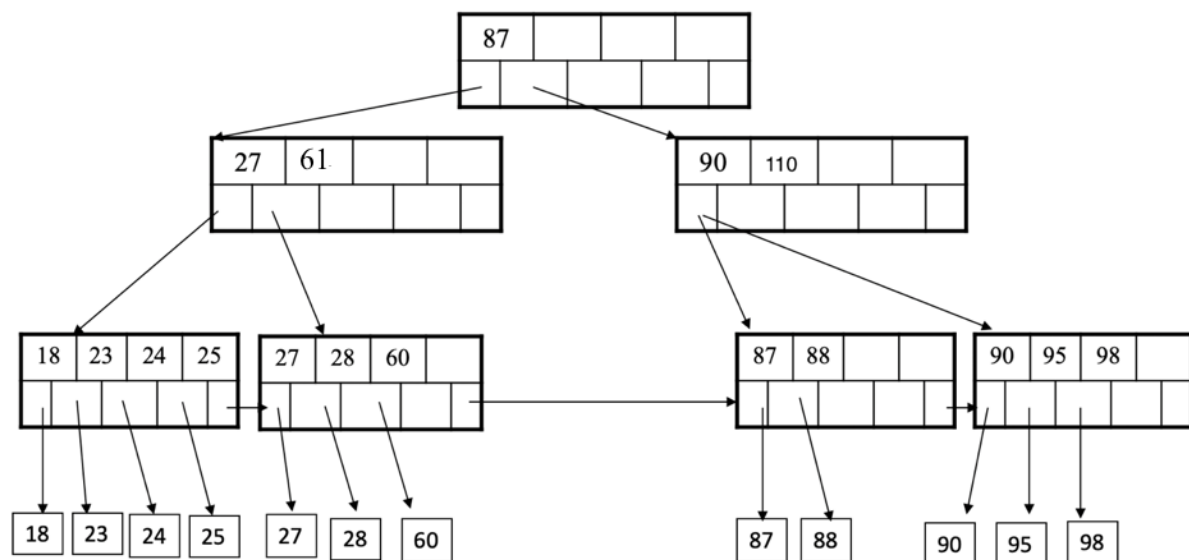
After inserted 40



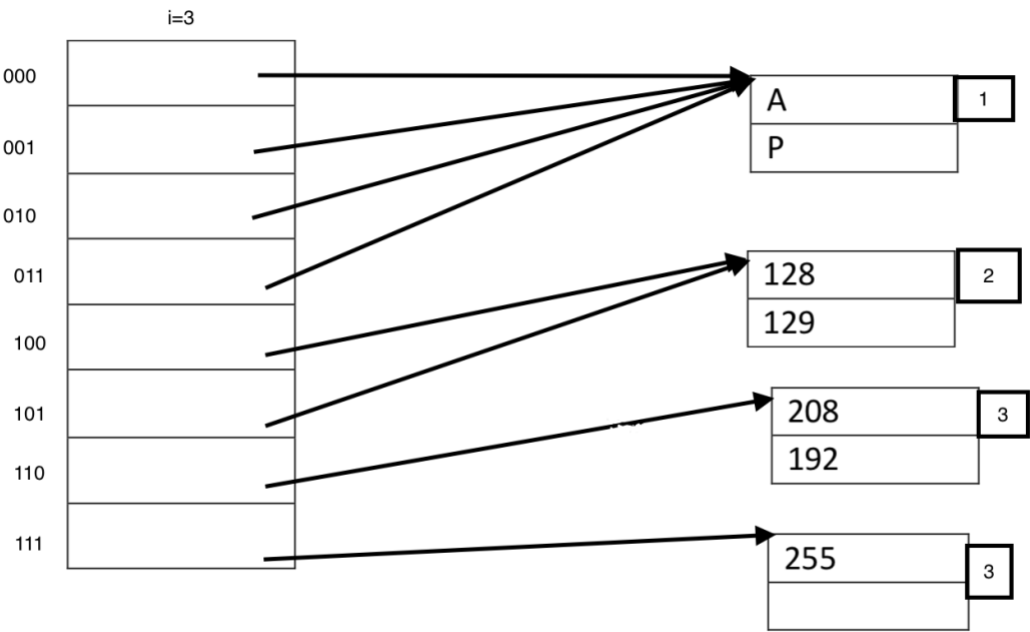
b)



After deleted 40



2.



3.

a) b)

Query	Attribute	Clustered /Not clustered	Index
Select * From Emp, Works Where 1 < Emp.eid < 200 and salary < 20000 and Emp.eid = Works.eid	Emp.eid	Non clustered	B+ tree
	Emp.salary	clustered	
	Works.eid	Non clustered	
Select * From Emp, Works, Dept Where Emp.eid = Works.eid and Works.did = Dept.did and budget= 80 0000	Emp.eid	clustered	Hash
	Emp.did	Non clustered	
	Work.eid	Non clustered	
	Work.did	Non clustered	
	Dept.budget	Non clustered	

4

a)

if using the nested loop join

Memory requirement: M

Cost: $B(R) + [B(R) / (M - 2)] B(S)$

almost $B(R) B(S) / M = (80\ 000 * 20\ 000) / 120\ 000 = 13\ 333.33$

The nested loop is the fastest way. however, the question said that it wants the output of join sorted according to attribute A. Therefore, use two passes soft merge join

b)

the buffer block size is small, so we consider to using the two passes soft merge join

Memory Requirement: $B(R) + B(S) \leq M^2$

Cost: $3B(R) + 3B(S) = 300\,000$

5.

If using optimized sort-merge join algorithm

The memory should be $B(R) + B(S) \leq M^2$

$80\,000 + 20\,000 \geq 10^2$

Therefore, we could change the merge time in order to expand the M size. As you can see, we could add another merge 3 time after one sorting and one merge, which will be $10^2 \times 10^2 \times 10^2$

$80\,000 + 20\,000 = 10^5$

Number of passes $\lceil \log_{M-1} \lceil B(R)+B(S) / M \rceil \rceil + 1 = 6$

After adding the sort and join

Cost : $11B(R) + 11B(S) = 1\,100\,000$