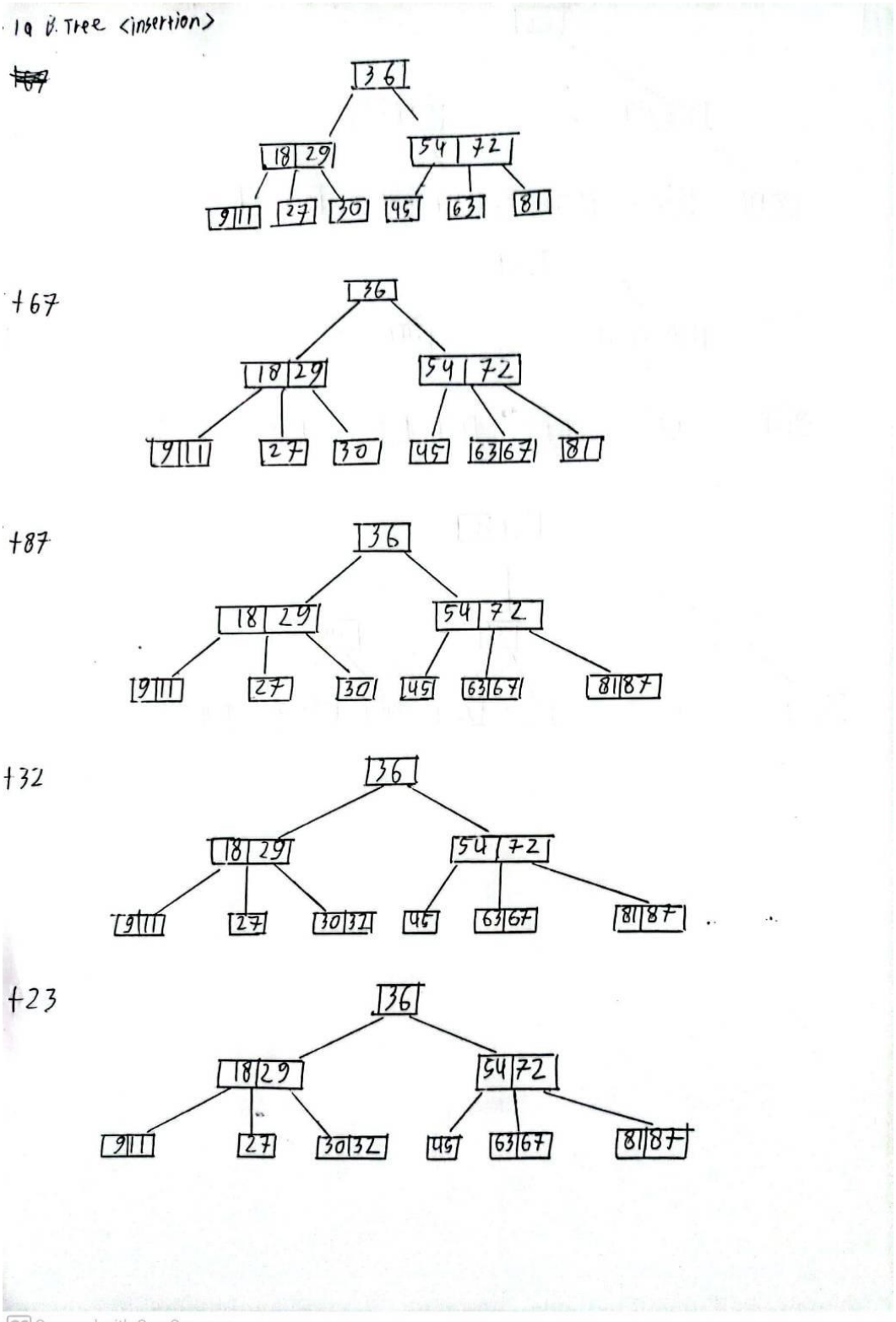
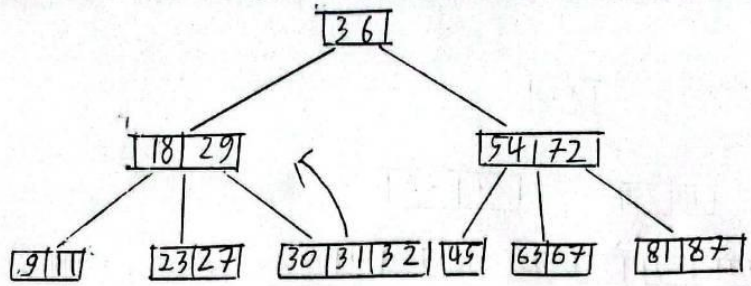
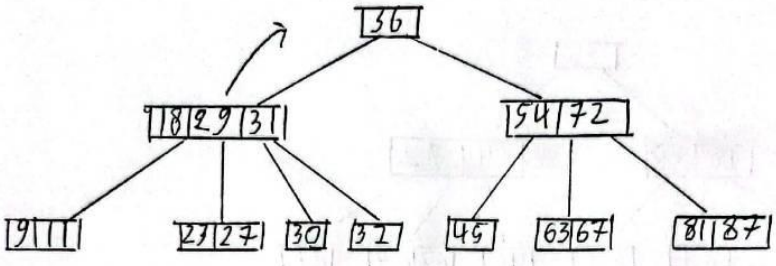
Steven - 2540125371

1a.

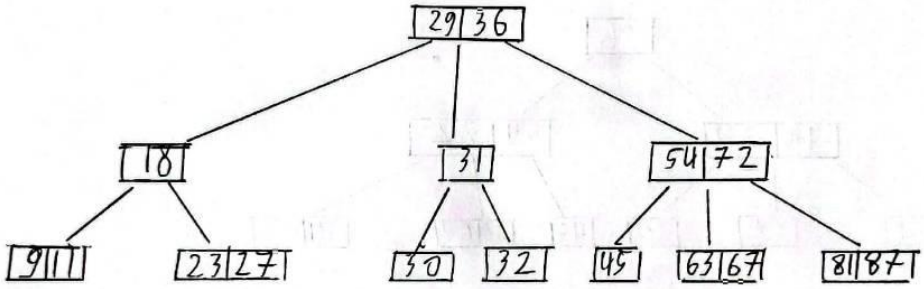


+31

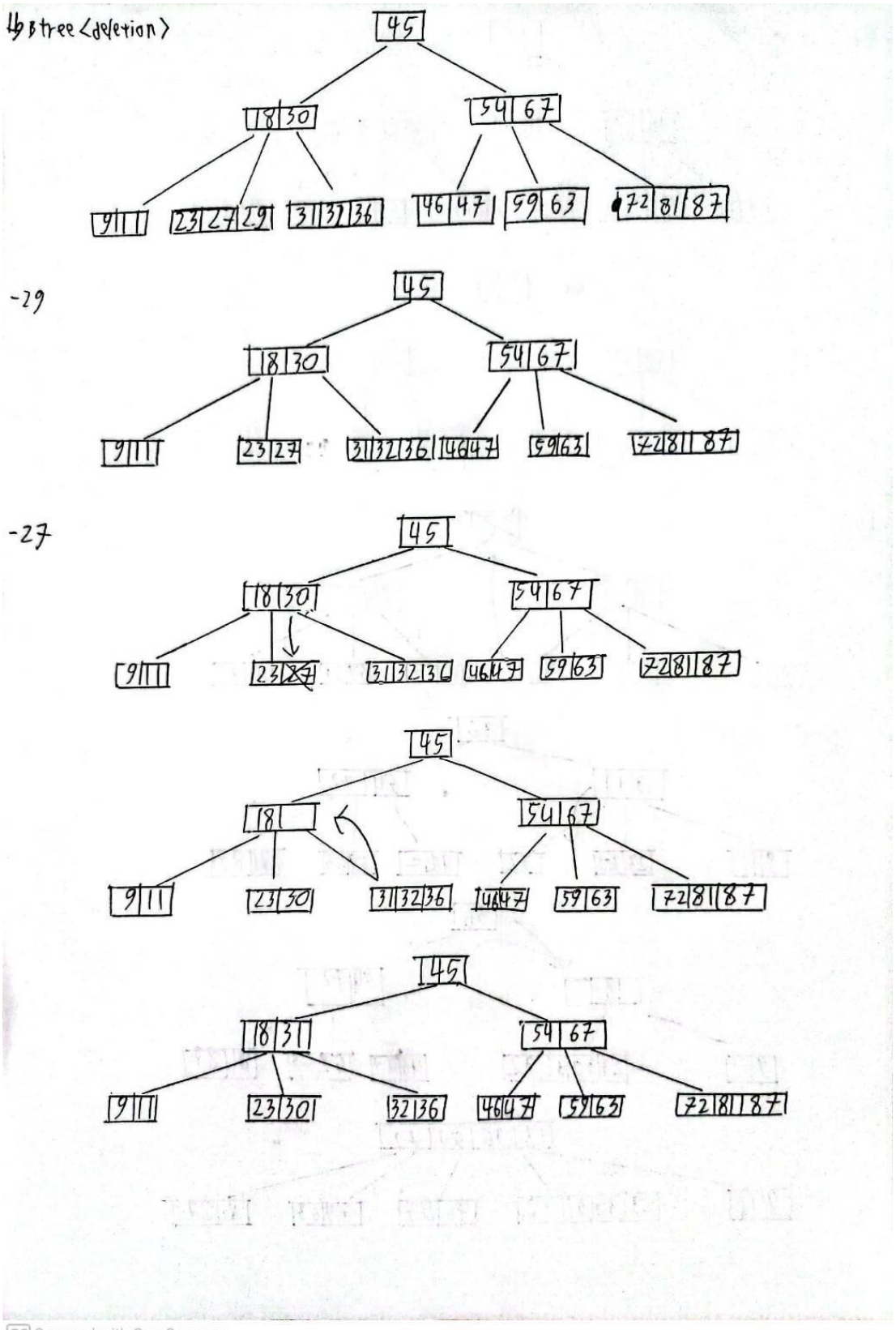




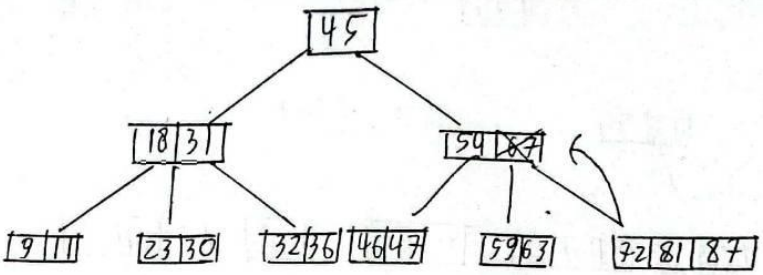
2

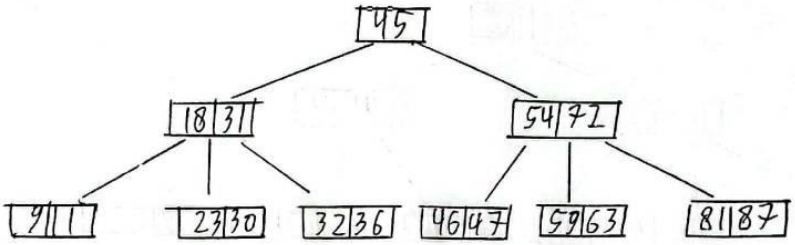


1b.

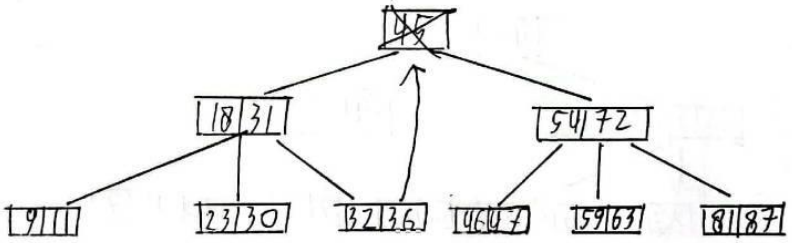


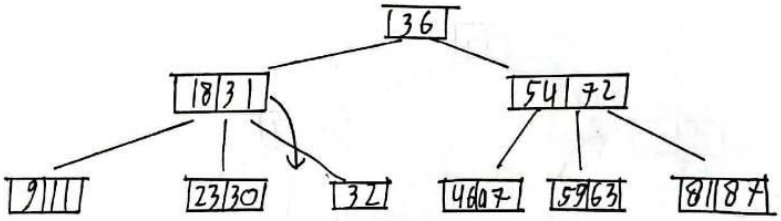
2. -67

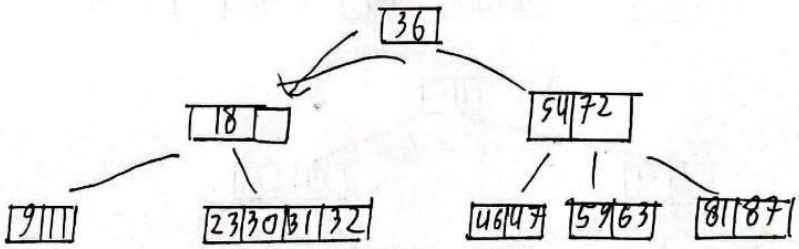


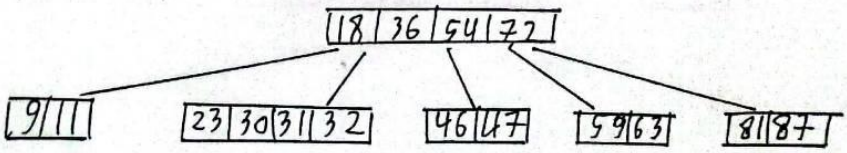


-45

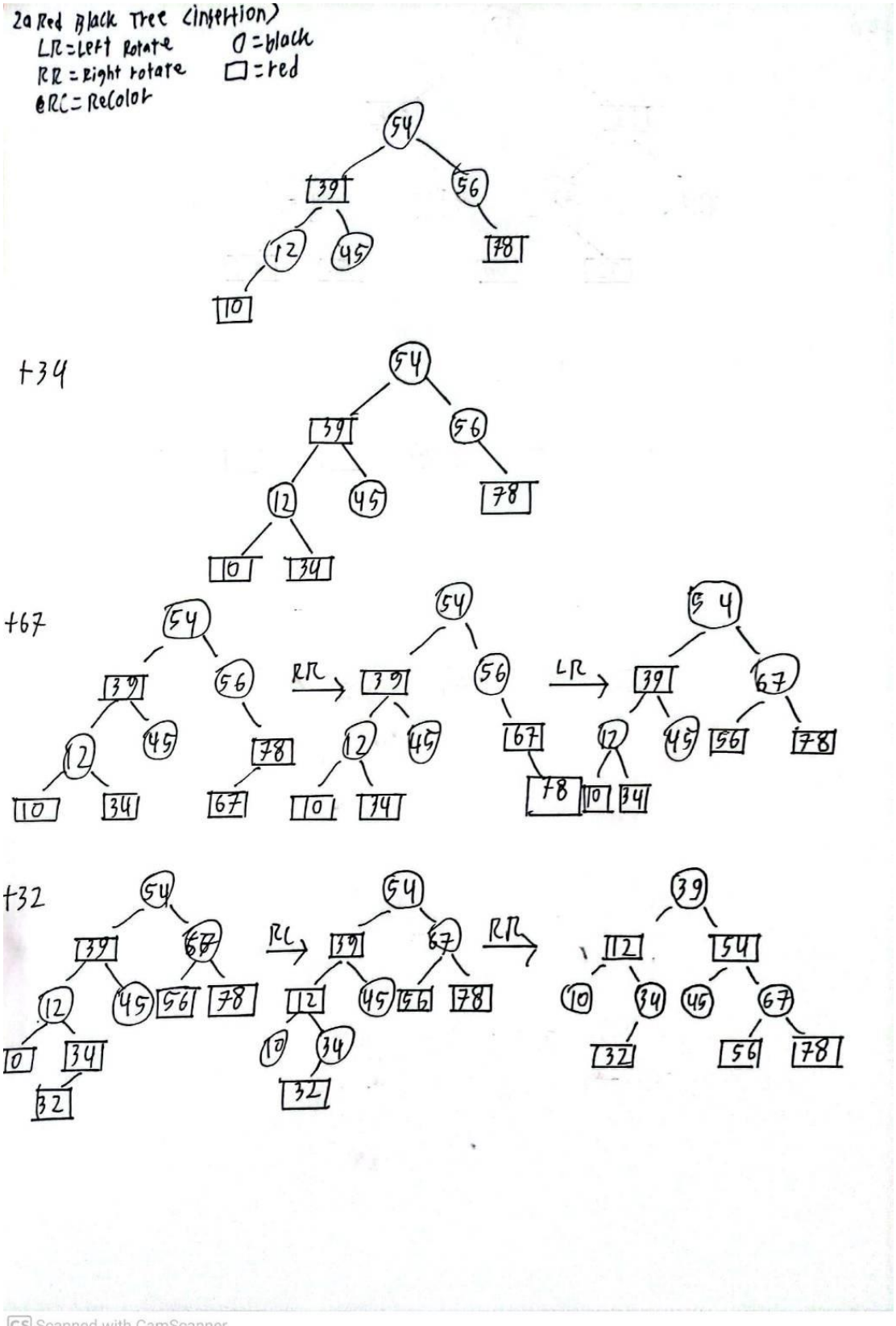




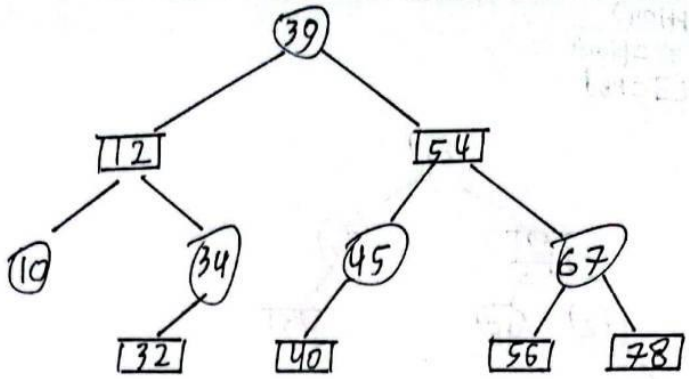




2a.

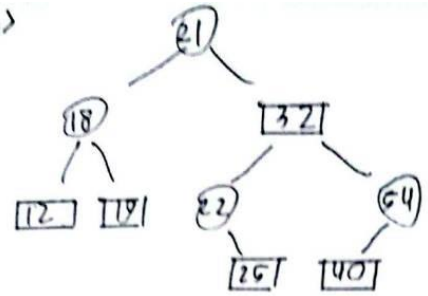


+40

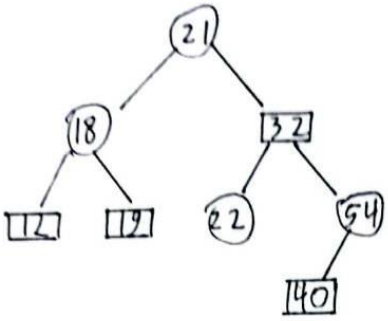


2b.

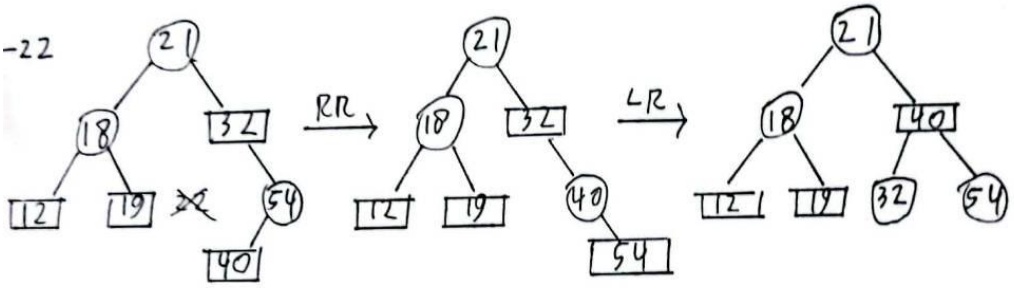
2b RB Black tree <deletion>
LR=Left Rotate
RR=Right rotate
RC=Recolor



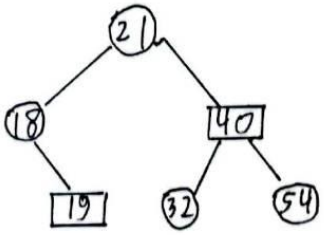
-25



-22



-12



3.

```c
#include<stdio.h>
#include<limits.h>
#define MAX 50
#define N 5 //number of Nodes
#define S 0 //starting vertex

void fixMatrix(int adjList[MAX][MAX]){
    for(int i=0;i<N;i++)
        for(int j=0;j<N;j++)
            if(adjList[i][j]==0)adjList[i][j]=INT_MAX;
}

int min_value(int d[], int visited_value[]){
    int smallest = INT_MAX;
    int smallest_idx;

    for(int i = 0; i < N; i++){
        if(visited_value[i] == 0 && d[i] < smallest){
            smallest = d[i];
            smallest_idx = i;
        }
    }

    return smallest_idx;
}

void findMST(int start, int adjList[MAX][MAX], int mst[MAX][MAX])
{
    int visited[MAX],d[MAX],parent[MAX];
    int min,u,v;

    for(int i=0;i<N;i++){
        d[i]=adjList[start][i];
        visited[i]=0;
        parent[i]=start;
    }

    visited[start]=1;

    int k=0;
    for(int i=0;i<N-1;i++){

        min=INT_MAX;
```

```
        //a. complete this line to select edge that has minimum weight (10)
        u = min_value(d, visited);

        visited[u]=1;

        mst[k][0]=parent[u];
        mst[k][1]=u;
        k++;

        for(v=0;v<N;v++)
            if(visited[v]==0 && (adjList[u][v]<d[v])){
                d[v]=adjList[u][v];
                parent[v]=u;
            }
    }
}


int main(){
    int adjList[MAX][MAX]={
    //b. complete this line to Initialize adjacency matrix for Graph given (5)
    {0, 3, 0, 1, 0},
    {3, 0, 2, 7, 9},
    {0, 2, 0, 0, 1},
    {1, 7, 0, 0, 5},
    {0, 9, 1, 5, 0}
    };

    int mst[MAX][MAX];
    fixMatrix(adjList);
    findMST(S,adjList,mst);

    printf("Edges of MST:\n\n");
    printf("Edge => Weight\n");
    printf("==============\n");

    int cost=0;
    for(int i=0;i<N-1;i++){
        int v1=mst[i][0];
        int v2=mst[i][1];
        printf("%2d%2d => %d\n",v1,v2,adjList[v1][v2]);

        //c. complete this line to compute total cost of MST(5)
        cost += adjList[v1][v2];
```

```
    }
    printf("==============\n");
    printf("Total Cost: %d\n",cost);

return 0;
}
```

4. di file cpp