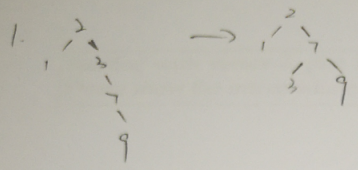
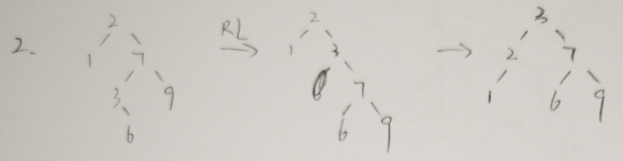
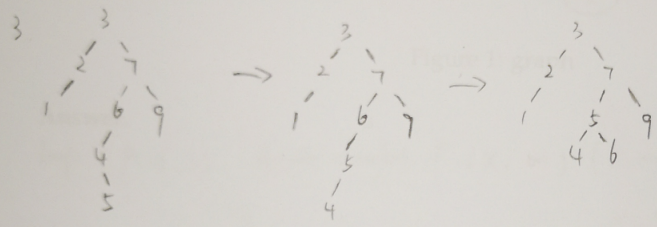
1.



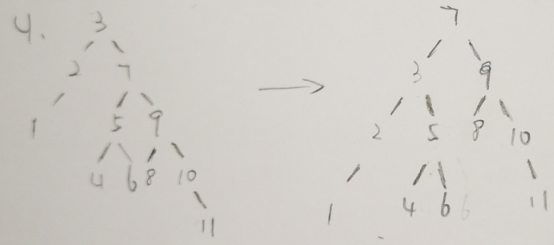
2.



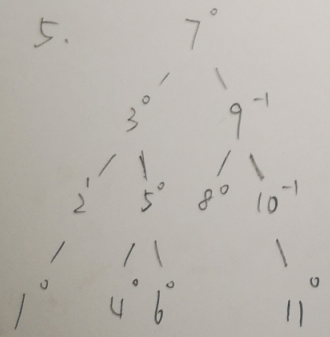
3



4.



5.



Question 2

Starting with vertex 7, plot the MST for the graph above using Prim's algorithm. Please show the intermediate steps (including 8 MST graphs).

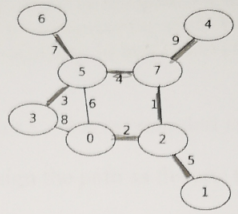
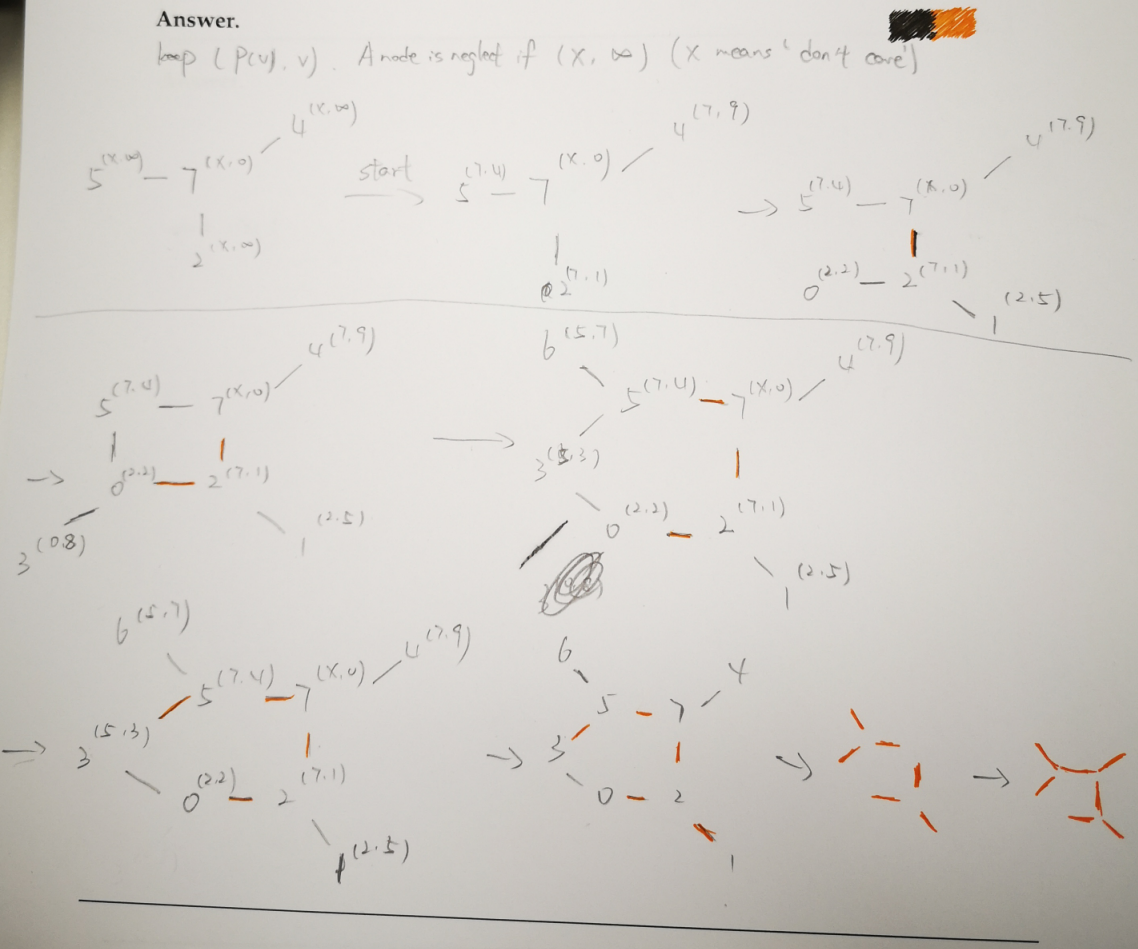


Figure 1: graph

Answer.



Question 3. Safe the Blue Tiger!

The blue tiger is stuck in a $m \times n$ grid, where there is non-negative number in each cell of the grid. Notice that the blue tiger **can only** move down or left in this game. In this homework, we are going to apply what we learnt and find a path from the top right of the grid to the bottom left of the grid. We should sum up all the numbers along the path and our goal is to minimize the sum. We need the time complexity to be $\mathcal{O}(n \times m)$ and the space complexity to be $\mathcal{O}(\min(m, n))$. You only need to write down the psudeo code.

Sample: Assume the grid of size 3×4 is represented as

8 8 1 1
8 8 1 8
1 1 1 8

The minimum sum is 6, by design the path as the one that cross all the 1s in the grid.

Template:

```
int solution(const int grid[][]){
    //To do
}
```

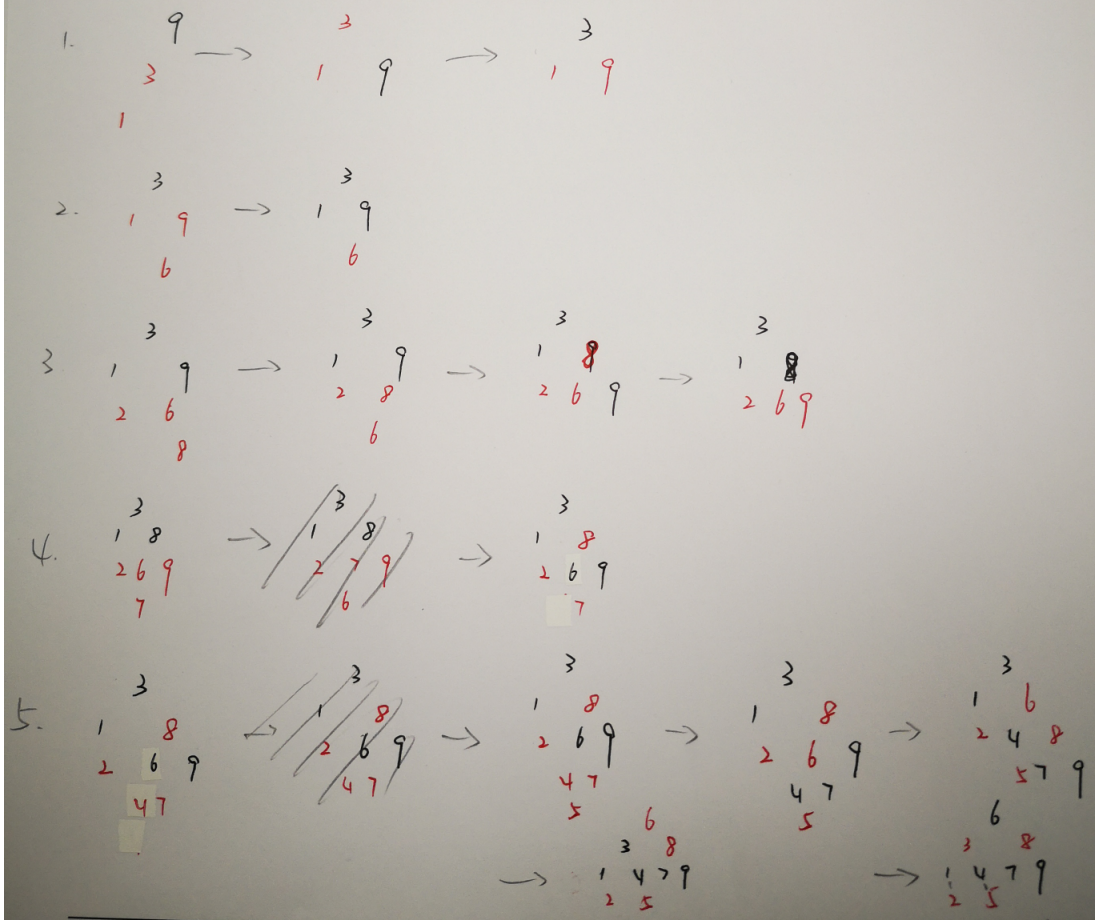
Answer.

```
int solution (const int grid[][]) {
    int [] dp = new int [m]
    for int (i=m-1; i>0; i--){
      dp = dp[0]+ grid[i][0]
      for (int j= ; j<n-1; j++) {
        dp[j] =mymin ( dp[j-1], dp[j] + grid[i][j])   // if dp[j-1] do not exist,
                                                      min will be dp[j]
      }
}
return dp[grid[0][n-1]
```

Question 4. RB Tree

1. Suppose that we insert a sequence of keys 9, 3, 1 into an initially empty red-black tree. Draw the resulting red-black tree.
2. Suppose that we further insert key 6 into the red-black tree you get in Problem (1-a). Draw the resulting red-black tree.
3. Suppose that we further insert keys 2, 8 into the red-black tree you get in Problem (1-b). Draw the resulting red-black tree.
4. Suppose that we further insert key 7 into the red-black tree you get in Problem (1-c). Draw the resulting red-black tree.
5. Suppose that we further insert keys 4, 5 into the red-black tree you get in Problem (1-d). Draw the resulting red-black tree.

Answer:.



Answer:.

```
int [][] dp = new int[n][5]      // 5 state  initial, buy 1st, sell 1st, buy 2nd, sell 2nd
dp[0] = {0, -price[0], 0, -price[0], 0};
for (int i=1; i<n; i++) {
  dp[i][0] = dp[i-1][0];
  dp[i][1] = max(dp[i-1][1], dp[i-1][0] - price[i]);
  dp[i][2] = max(dp[i-1][2], dp[i-1][1] + price[i]);
  dp[i][3] = max(dp[i-1][3], dp[i-1][2] - price[i]);
  dp[i][4] = max(dp[i-1][4], dp[i-1][3] + price[i]);
}
return max(max(max(dp[n-1][0], dp[n-1][1]), max(dp[n-1][2], dp[n-1][3])), dp[n-1][4]);
```