30 Days Leetcode challenge - Day 23

public class leetday23

{

class Solution1 {

public int[] findPeakGrid(int[][] arr) {

int n = arr.length;

int m = arr[0].length;

int low = 0;

int high = m-1;

int left = -1;

int right = -1;

while(low<=high)

{

int mid = (low+high)/2;

int row = maxElement(arr,n,m,mid);

if(mid-1>=0) // int left = (mid-1>=0) ? arr[row][mid-1] : -1;

{

left = arr[row][mid-1];

}

else

{

left = -1;

}

if(mid+1<m)

{

right = arr[row][mid+1];

}

else

{

right = -1;

}

if(arr[row][mid]>left && arr[row][mid]>right)

{

return new int [] {row,mid};

}

else if(arr[row][mid]<left)

{

high = mid-1;

}

else

{

low = mid+1;

}

}

return new int [] {left,right};

}

public static int maxElement(int arr [] [], int n, int m, int column)

{

int max = -1;

int index = -1;

for(int i=0;i<n;i++)

{

if(arr[i][column]>max)

{

max = arr[i][column];

index = i;

}

}

return index;

}

}

class Solution2 {

public String removeOuterParentheses(String s) {

StringBuilder result = new StringBuilder();// to store the Output // 2.( // 3.()

int level = 0;

for(char c : s.toCharArray())

{

if(c=='(')

{

if(level>0)

{

result.append(c);

}

level++;

}

else

{

level--;

if(level>0)

{

result.append(c);

}

}

}

return result.toString();

}

}

}