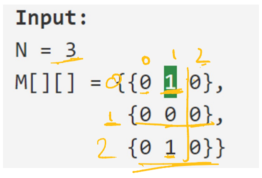
22-Jan-2022

22 January 2022 20:10

https://practice.geeksforgeeks.org/problems/the-celebrity-problem/1/#



```
Input:
N = 3
M[][] = {{0 1 0},
         {0 0 0},
         {0 1 0}}
```

3x3

0 to n-1

(1)[0)

[0](1)

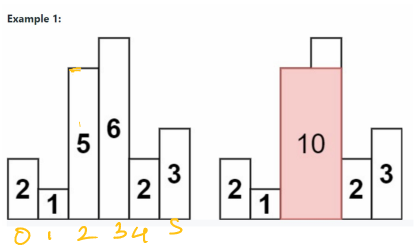
```
if  m[i][j] == 1
    i knows j

[1][0] == 0
while (! st.isEmpty) {
    j = st.pop()   // 2
    i = st.pop()   // 1
    if (m[i][j] == 1) {
        st.push(j);
    } else if (m[j][i] == 1) {
        st.push(i);
    }
}
```

Largest Histogram

https://leetcode.com/problems/largest-rectangle-in-histogram/

width of one bar = 1

Example 1:



6 2 5 4 5 1 6

0 1 2 3 4 5 6
6
2
5
4
5
1
6

8
6
12

LSI = 1 2 1 2 1 6 1

RSI = 1 4 1 2 1 2 1

$ar[i] \times (LSI[i] + RSI[i] - 1)$

LSI = -1, -1

RSI = 1 2

i = 1

$12 = ar[i] \times (RSI[i] + LSI[i] + 1)$

0	6	(1	)	= 6
	2 ×	(5	)	= 10
	5 ×	(1	)	= 5
	4 ×	(3	)	= 12
	5 ×	(1	)	= 5
	1 ×	(7)		= 7
	6 ×	(1)		= 6

LSI

i = 0 1 2 3 4 5 6

6 2 5 4 5 1 6

1 2 1 2 1 6 1

1 2 1 2 1 6 1

~~1~~
~~2~~
~~3~~
~~4~~
5
~~6~~

st.pop → i

6 5

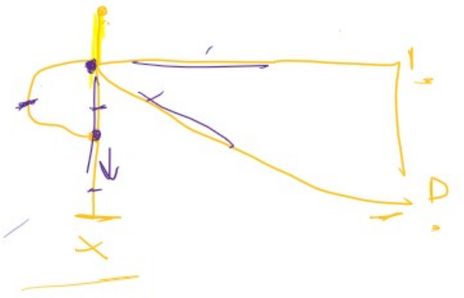
0	1	2	3	4	5	6
6	2	5	4	5	1	6
1	4	1	2	1	2	1

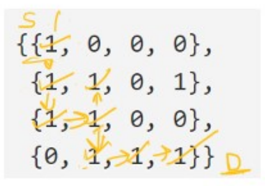
R S I

6 –
5 –
~~4~~
~~3~~
~~2~~
~~1~~
~~0~~

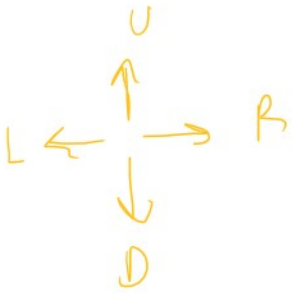
an. len – S

Backtracking:-



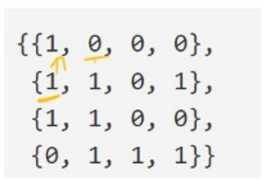


```
{{1, 0, 0, 0},
 {1, 1, 0, 1},
 {1, 1, 0, 0},
 {0, 1, 1, 1}}
```



D R D D R R

D D R D R R



```
{{1, 0, 0, 0},
 {1, 1, 0, 1},
 {1, 1, 0, 0},
 {0, 1, 1, 1}}
```

[

```
findPath(arr, r, c, path)  list.add(path); return;
if(r == n-1 && c == n-1) {
arr[r][c] = 0
if(r-1>=0 && arr[r-1][c]==1){
    findPath(arr, r-1, c, path+"U")
}
if(r-1>=0 && arr[r-1][c]==1){
        findPaths(arr, r-1, c, path+"U");
}
if(c-1>=0 && arr[r][c-1]==1){
        findPaths(arr, r, c-1, path+"L");
}
if(r+1<n  && arr[r+1][c]==1){
        findPaths(arr, r+1, c, path+"D");
}
if(c+1<n  && arr[r][c+1]==1){
        findPaths(arr, r, c+1, path+"R");
}
arr[r][c]=1
}
```