5/10/2023

Graphs - 5

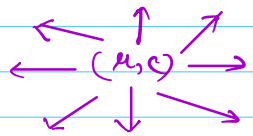
2 weeks

1> Schedule mock interview
2> Complete assignments & get to PsP ≥ 85
3> Contest re-attempt.

Q1 Given a matrix of integers with 1 & 0 in each cell.

1 → land 0 → water

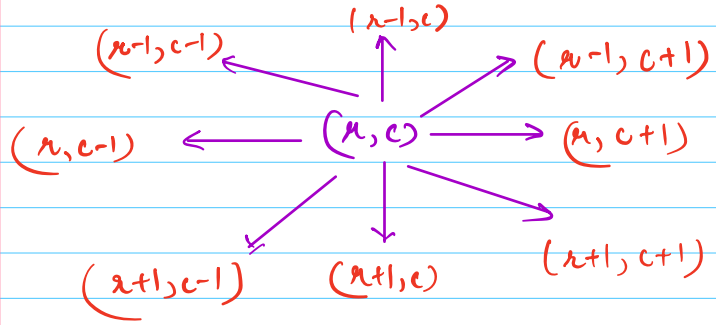
A set of connected 1's → island.
Find the # island(s) in the matrix.
(8 directions & not 4 directions)



$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Ans = 4

Ans = # connected components.



	0	1	2	3	4	5	6	7
dr =	-1	-1	-1	0	0	+1	+1	+1
dc =	-1	0	+1	-1	+1	-1	0	+1

```
ans = 0 , if i,j, vst[i][j] = false
for i -> 0 to (N-1){
    for j->0 to (M-1){
        if (!vst[i][j] && A[i][j] == 1){
            ans++;
            dfs(i,j)
        }
    }
}
```

```
void dfs (r,c) {
    vst[r][c] = true;
    for i -> 0 to 7/dr.length{
        x = r + dr[i]
        y = c + dc[i]
        if (x>=0 && x<N && y>=0 && y<M &&
            !vst[x][y] && A[x][y]==1)
            dfs(x,y);
    }
}
```

TC: O(N*M)
SC: O(N*M) → 2xNxM
- → vst[][]
- → stack

Q2 Given an array of positive elements, flip sign of some of its elements (* -1) s.t. sum of elements of the final array is min non-negative integer (>= 0)

Find minimum # elements to flip to achieve this task.

A = [10 ~~15~~ 6 ~~3~~ 3] → 10 - 15 + 6 - 3 + 3
-15 -3
=> 1, Ans = 2

A[i] → flip (take) or not-flip (leave)

Sum of elements flipped <= Sum of elements not flipped.

Total Sum = S
Sum of elements flipped <= S/2

Ans = min # elements flipped.

Find min # elements to flip s.t. sum of selected elements <= S/2

Bag capacity ⟶ S/2 (Total Sum/2)

Loss ⟶ 1 per element.

weight of i^{th} elements ⟶ A[i]

1) Matrix exponentiation
2) digit dp
3) Binary lifting
4) BIT
5) Segment tree

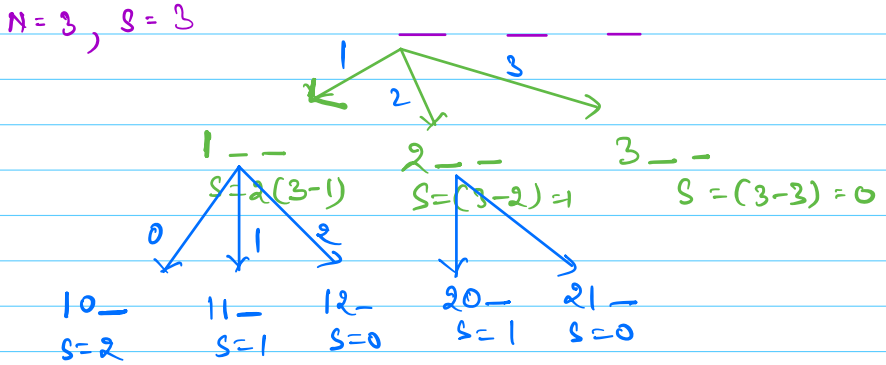
digit dp

Q3 Find the count of N digit no whose digit sum = S (>0). Leading 0's are not allowed. Eg 012X

N=2, S=4 { 13, 22, 31, 40 }, Ans = 4

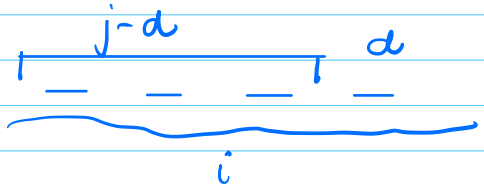
N=2, S=3 { 12, 21, 30 }, Ans = 3

N = 3, S = 3



count[i][j] → # i digit nos with digit sum j

Last step



$$count[i][j] = \sum_{d} count[i-1][j-d]$$

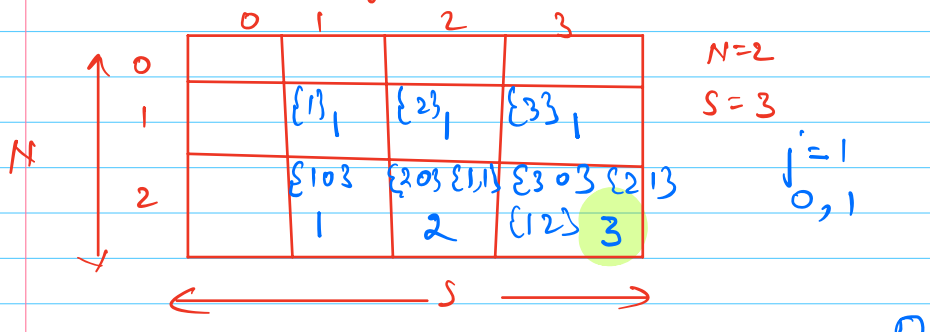
If i, j count[i][j] = 0

```
for i -> 1 to N {
    for j -> 1 to S {
        if (i == 1) {
            if (j <= 9) count[i][j] = 1
            else count[i][j] = 0;
        }
        else {
            for d -> 0 to 9 {
                if (j >= d) {
                    count[i][j] += count[i-1][j-d]
                }
            }
        }
    }
}
```

TC : O(N * S * 10) ≃ (N × S)

SC : O(N * S)

return count[N][S]



N = 2

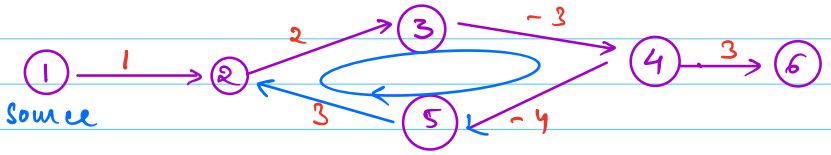
S = 3

j = 1
0, 1

Q

Break till 8:38 am IST

Q4 Is it always possible to find single source shortest path with -ve weights.



shortest path from 1 → 6 ?

1 → 2 → 3 → 4 → 6 (3)

1 → 2 → 3 → 4 → 5 → 2 → 3 → 4 → 6 (1)

Negative wt. cycle → cycle of nodes s.t. sum of edge weights < 0

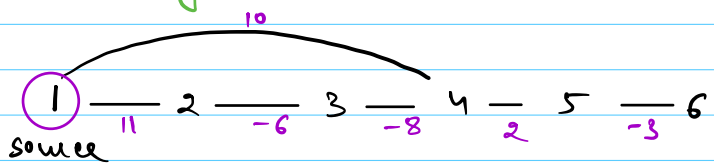
Q 5 Find single source shortest path (-ve wt possible)
No -ve weight cycle in the graph.

Bellman ford algo

Minimum distance can be found by updating/relaxing all edges (N-1) times, irrespective of the order in which edges are selected.

max # edges b/w 2 nodes

if (d[u] + wt(u,v) < d[v])
d[v] = d[u] + wt(u,v)



d = [0 ~~11~~ ~~5~~ ~~10~~ ~~12~~ ~~4~~]
-3 -1 -4
1 2 3 4 5 6

	1 to 1	2	3	4	5
5 —(-3)— 6	X	X	✓	X	✓
4 —(2)— 5	X	✓	X	✓	X
3 —(-8)— 4	X	X	✓	X	X
2 —(-6)— 3	X	✓	X	X	X
1 —(10)— 4	✓	X	X	X	X
1 —(11)— 2	✓	X	X	X	X

p = [-1 1 2 3 4 5]
1 2 3 4 5 6

1 — 4
d[1] + wt(1,4) < d(4)
0 + 10 < ∞
d[4] = d[1] + wt(1,4)
= 0 + 10

```
∀ i   d[i] = INT_MAX
  d[1] = 0

for  i -> 1 to (N-1) {
      stop = true;

      for ( edge (u,v) : Edges) {

          if (d[u] + wt[u v] < d[v]){
                d[v] = d[u] + wt(u,v)
                 stop = false;
                pre[v] = u;
          }
      }

      if (stop == true) break;
}
return d;
```

Tc: O(N*E)
Sc: O(1) / O(N)