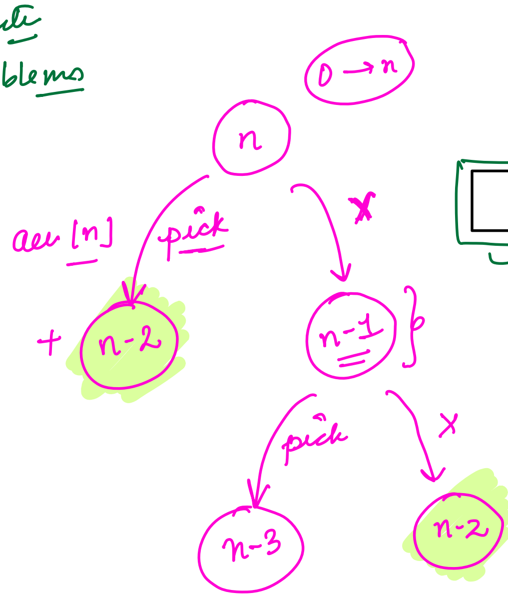
Q Given an array, Find max subsequence sum.
adjacent elements

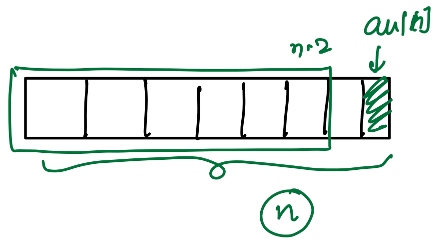
9 4 13

8 -1 -3 -2 5

9 4 13 24

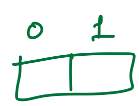
optimal substructure
overlapping subproblems





$$\text{max sum}(n) = \max(\text{arr}[n] + \text{maxsum}(n-2), \text{maxsum}(n-1))$$

~~dp~~ int maxSum[n];



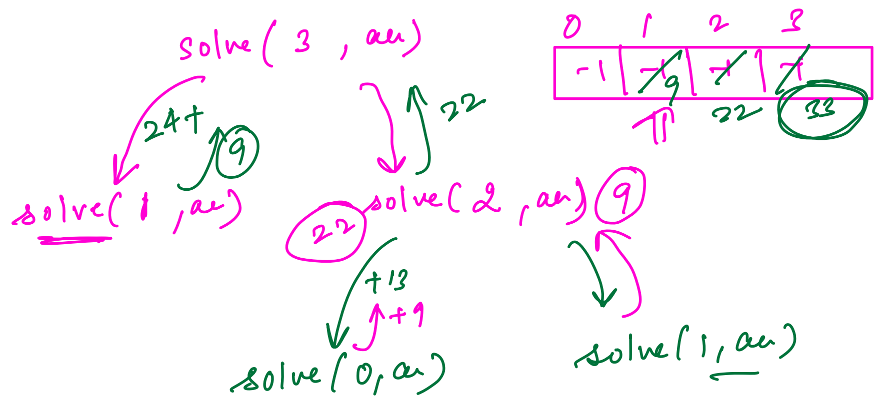

```
solve( int n, int arr)
{
    if (n==0) return max(0, arr[n]);
    if (n==1) return max(0, arr[0], arr[1]);

    if( maxsum[n] != -1) return maxsum[n];
```

maxsum[n] = max(arr[n] + solve(n-2, arr), solve(n-1, arr))

return maxsum[n];

}

0	1	2	~~0~~ 3
9	4	13	24



9	4	13	24	19	6	41
9	9	22	33	41	6+33	
		13+9	~~24+9~~	19+22	41	
		9	22	33		

T.C: O(n)

S.C: O(n) iterative - O(1)

- 2 * N matrix

$$\begin{array}{cccccc} 4 & 1 & 8 & 5 & 3 & 8 \\ \end{array}$$
$$\begin{bmatrix} 2 & -3 & 8 & 5 & 1 & 6 \\ 4 & 1 & 5 & 2 & 3 & 8 \end{bmatrix}$$

adjacent
hz ✗
vertical ✗
diagonal.

If you decide to pick an element from i^{th} column
$i-1$ ✗ $i+1$ ✗

array[i] = max(matrix[0][i], matrix[1][i]);

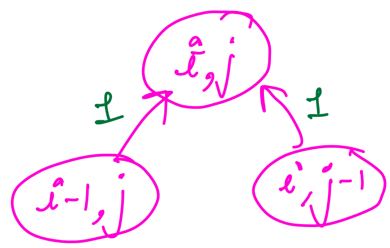
Q

	0	1	2	3
0	1	1	1	1
1	1	2	3	
2	1			
3	1			
4	1			

n*m

no of ways to go from (0,0) to (n-1)*(m-1)

right →, down ↓ } only allow



ways(i,j) = ways(i-1,j) + ways(i,j-1)

$$\boxed{dp[i][j] = dp[i-1][j] + dp[i][j-1]}$$

if(i==0 || j==0) dp[i][j]=1;

Pune → Banglore ← HYD

iterative - T.C: O(n*m)

S.C: O(n*m)

but if you optimise

→ O(2*m)

Because you only need last 2 rows

Problem solving: dp session

	0	1	2	3
0	1	1	1	1
1	1	0	1	0
2	0	1	1	1
3	1	0	1	1
4	1	1	1	1

no of ways $(0,0) \rightarrow (n-1, m-1)$

$mat[i][j] = 0 \Rightarrow$ blocker

right →

↓ down

	0	1	2	3
0	1 1	1 1	1 1	1 1
1	1 1	0 0	1 1	0 0
2	0 0	1 0	1 1	1 1
3	1 0	0 0	1 1	1 2
4	1 0	1 0	1 1	1 3

```
if (mat[i][j] == 0)
        dp[i][j] = 0;
else
    dp[i][j] = ________;
```

#

	0	1	2	3	4
0	2 2	1 3	3 6	7 13	3 16
1	3 5	3 6	4 10	8 18	3 19
2	0 5	5 10	1 11	2 13	1 14
3	0 5	2 7	3 10	1 11	3 14
4	4 9	1 8	5 13	2 13	3 16

To enter a particular cell, you have to pay some cost

min cost $\rightarrow (0,0) \rightarrow (n-1, m-1)$

$$mincost(i, j) = cost[i][j] + min(mincost(i, j-1), mincost(i-1, j));$$

Dragons & princess / prince

	0	1	2	3
0	-3	2	4	-5
1	-6	5	-4	6
2	-15	-7	5	-2
3	2	10	-3	-4

→ Q

chamber → dragon
chamber → redbull.

→ ↓

≤ 0 energy → die

How much min energy should prince start with?

$\boxed{-4}$ ↓ x

$x + (-4) = 1$

$x = 1 - (-4) = 5$

	0	1	2	3
0	-3	2	4	-5
1	-6	5	-4	6
2	-15	-7	5	-2 (7)
3	2	10	-3	-4 (5)

-5 (12)	-2 (7)
-3 (8)	-4

5 (2)	-2 (7)
-3 (8)	-4

$x + (-2) = 5$

$x = 7$

$x + (-5) = \min(7, 8)$

$x = 12$

$x + 5 = \min(7, 8)$

$x = 7 - 5$

$= 2$

$x + arr[i][j] = \min(\_, \_)$

$$\boxed{dp[i][j] = \min(dp[i+1][j],\ dp[i][j+1]) - arr[i][j]}$$

$\downarrow x \; 1$

$\boxed{4}$

$x + 4 = 1$

$x = -3$

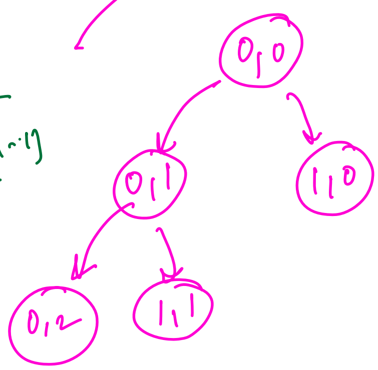
$$\boxed{dp[i][j] = \max\left(1,\ \min(dp[i+1][j],\ dp[i][j+1]) - arr[i][j]\right)}$$

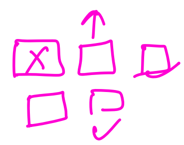
$$dp[n-1][m-1] = \max(1 - arr[n-1, m-1],\ 1);$$

$\downarrow x$ $\square \rightarrow$ last cell

$x + arr[n-1][m-1] = 1$

$x = 1 - arr[n-1][m-1]$





solve(0, 0)
　　　i, j

```
{
    if (i==n-1 && j==m-1)  ——→
    if (dp[i][j] != -1)  ——
    dp[i][j] = min( solve(i+1, j),
                    solve(i, j+1)) - arr[i][j]
```