DP Concepts & Questions

Video 35

If you r ally want something, you wo d cross every limit to get it.

{ Ask ourself, are you eally willing?? }

MIK...

भाषण
(Motivation)

cswithMIK → Twitter

Facebook] → code torywithMIK
Instagram]

whatsapp → codesl withMIK

Done
- 1-D based DP

- Grid based DP

Done
- String based DP

- Digit DP

- Game Strategy

We'll do :-

(•) RECURSION
+
MEMOIZATION
(Top Down)

(•) Bottom Up

(•) Time & Space

DP on Grids

3363. Find the Maximum Number of Fruits Collected

Hard | Topics | Companies | Hint

There is a game dungeon comprised of `n x n` rooms arranged in a grid.

You are given a 2D array `fruits` of size `n x n`, where `fruits[i][j]` represents the number of fruits in the room `(i, j)`. Three children will play in the game dungeon, with **initial** positions at the corner rooms `(0, 0)`, `(0, n - 1)`, and `(n - 1, 0)`.

The children will make **exactly** `n - 1` moves according to the following rules to reach the room `(n - 1, n - 1)`:

- The child starting from `(0, 0)` must move from their current room `(i, j)` to one of the rooms `(i + 1, j + 1)`, `(i + 1, j)`, and `(i, j + 1)` if the target room exists.
- The child starting from `(0, n - 1)` must move from their current room `(i, j)` to one of the rooms `(i + 1, j - 1)`, `(i + 1, j)`, and `(i + 1, j + 1)` if the target room exists.
- The child starting from `(n - 1, 0)` must move from their current room `(i, j)` to one of the rooms `(i - 1, j + 1)`, `(i, j + 1)`, and `(i + 1, j + 1)` if the target room exists.

When a child enters a room, they will collect all the fruits there. If two or more children enter the same room, only one child will collect the fruits, and the room will be emptied after they leave.

Return the **maximum** number of fruits the children can collect from the dungeon.

Ch1 → 0 ← Ch2

Example :- fruits =

	0	1	2	3
0	1	2	3	4
1	5	6	8	7
2	9	10	11	12
3	13	14	15	16

$n = 4$

$(n-1)$ moves $= 3$ moves

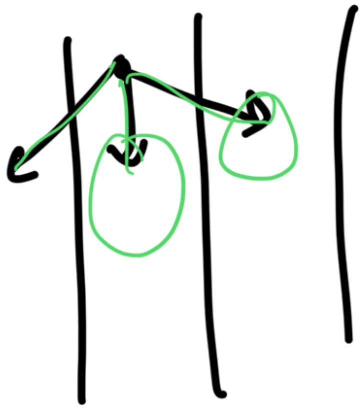
4x4

Output: $\boxed{1+6+11+16} + \; +8+12 + \boxed{13+14+15} = 100$

Thought Process

option (Recursion)

Minimum/Maximum Falling Path sum



⟹ minimum/su.

	0	1	2	3
Child1 0	1	2	3	4
1	5	6	8	7
2	9	10	11	12
3	13	14	15	16

n*n

$n = 4$

moves = 3

(n-1) moves.

★ Child-1 : Add the diagonal Elements.

C1 :

* Child - 2 : (Recursion) + constraint.

	0	1	2	
0				C2
1	(i,j)			
2	(i,j)	(i,j)		

$n = 4$

moves = (n-1) = 3

3 (i,j) | (i,j) | (i,j)

4*4

Constraint for Child 2 :

$\Leftarrow (i > j)$

&&

Child1 $\Leftarrow (i == j)$

(2,1)
(3,2)
(1,0)
(2,0)
(3,0)
(3,1)

$(i > j)$

$(0, n-1)$

$(i+1, j-1)$ $(i+1, j)$ $(i+1, j+1)$

Child - 3 : (Recursi + constraint)

	0	1	2	3
0				
1			?	?
2				

n = 4

moves = (n-1) = 3

constraint for C3 : $(i < j)$

||

$i == j)$ → child 1

Bottom Up :-

Recursion code :- two parameters were changing $\boxed{(i, j)}$ *

DP array = 2D matrix.

Ch1 → (n-1, n-1)
Ch2 →
Ch3 →

State Def :- $t[i][j]$ = maximum fruits collected when reache $[i][j]$ from source.

* Ch1 → diagonal ents sum.

fruits:-

1	2	10	4
5	6	8	7
9	10	11	12
13	14	15	16

t

	0	1	2	3
	1	~~4~~	~~4~~	4
1	5	6	8	7
	9	10	11	12
j	13	14	15	16

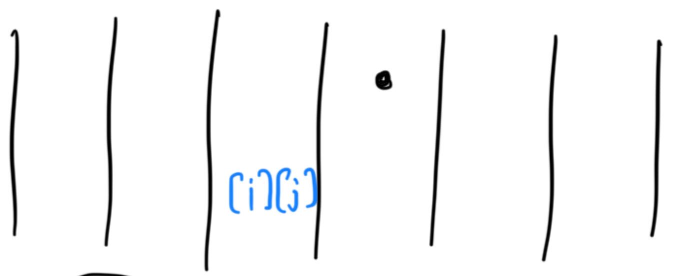
t[i][j] = max fruits collec till [i][j] from [0][n-1]

[1][2]

t[0][n-1] = 4

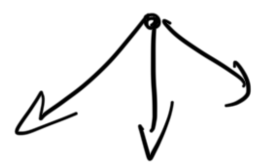
t[1][2] max fru colle till [1][2] from [0][n-1]

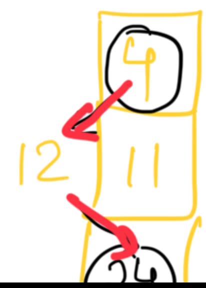
C2



C2 (i-1, j-1) (i-1, j) i-1, j+1

(i, j)





$t[i][j] =$

child 2:

$$t[i][j] \mathrel{+}= \max(t[i-1][j-1],\ t[i-1][j],\ t[i-1][j+1])$$

(0, n-1)

Ch2

```
for( i=1, i<n; i+ ){   (i<j)
    for( int j = i+1 ; j<n; j++){
        t[i][j] += max(t[i-1][j-1], t[i-1][j], t[i-1][j+1])
    }
}
```

i<j

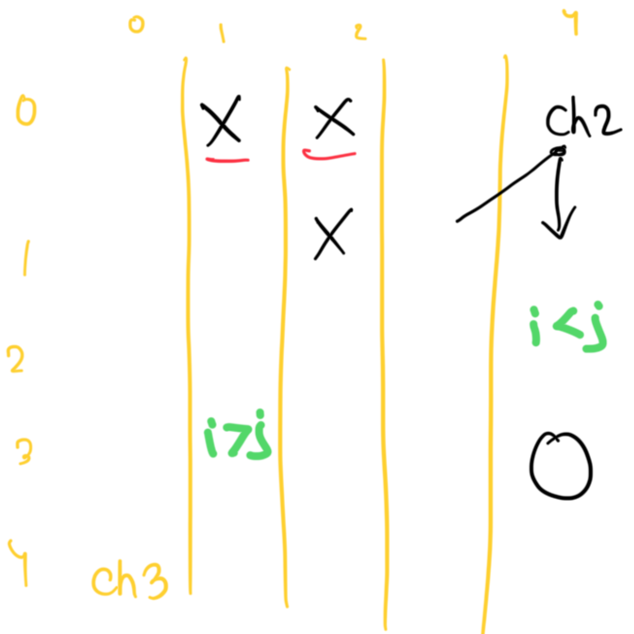
j+1<n

t[i-1][j+1]

0

n=5

n-1=4

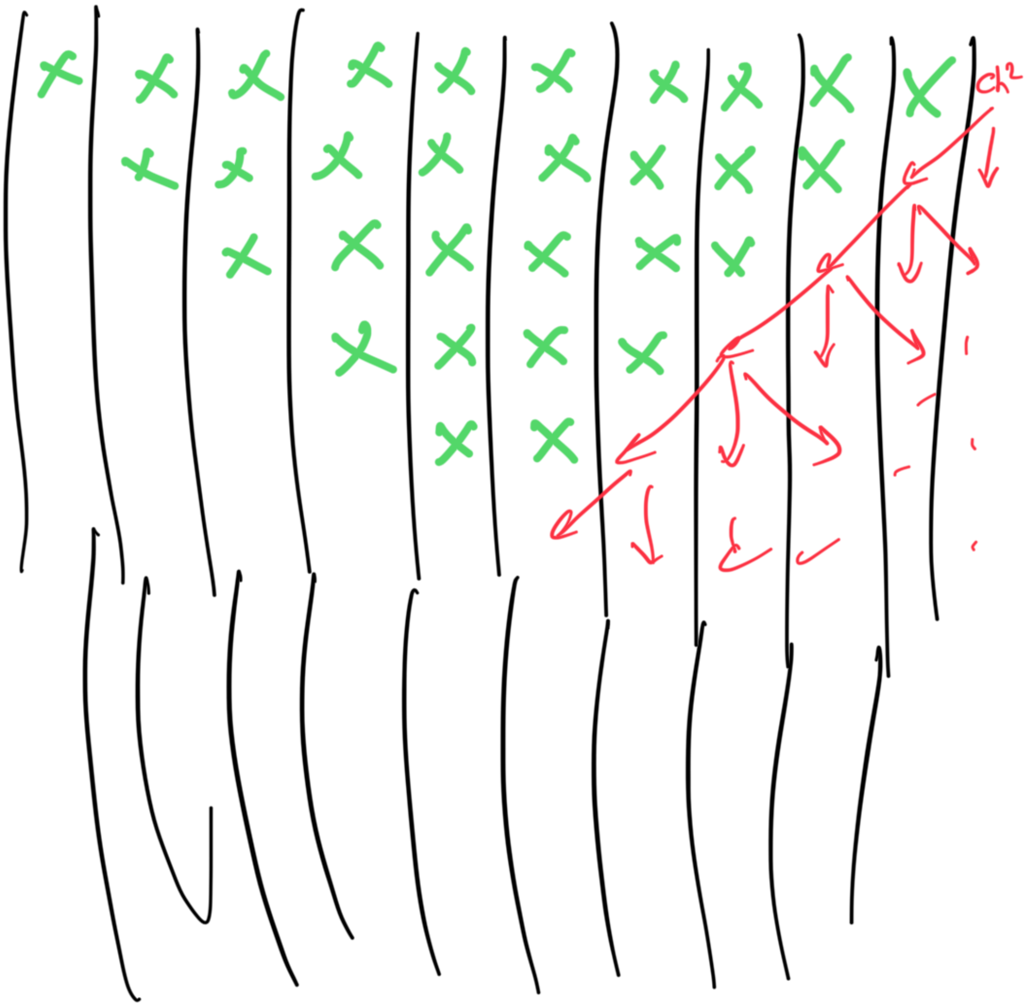


Ch2

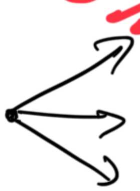
i j	i<j
(0,1)	0+1 < 4
(0,2)	0+2 < 4
(0,3)	0+3 < 4
(1,2)	1+2 < 4
(2,3)	2+3 < 4 X
(3,4)	3+4 < 4 X

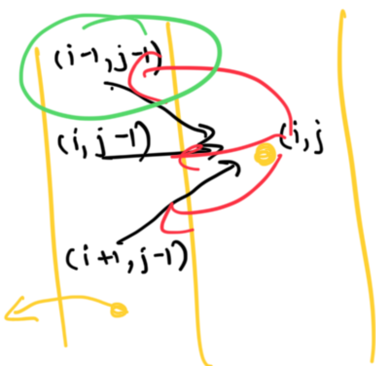
Ch-2 :- $i+j < n-1$

0



Ch-3





$i > j$

$(i > j)$

(i>j)

for(j = 1 ; j<n ; j++) {

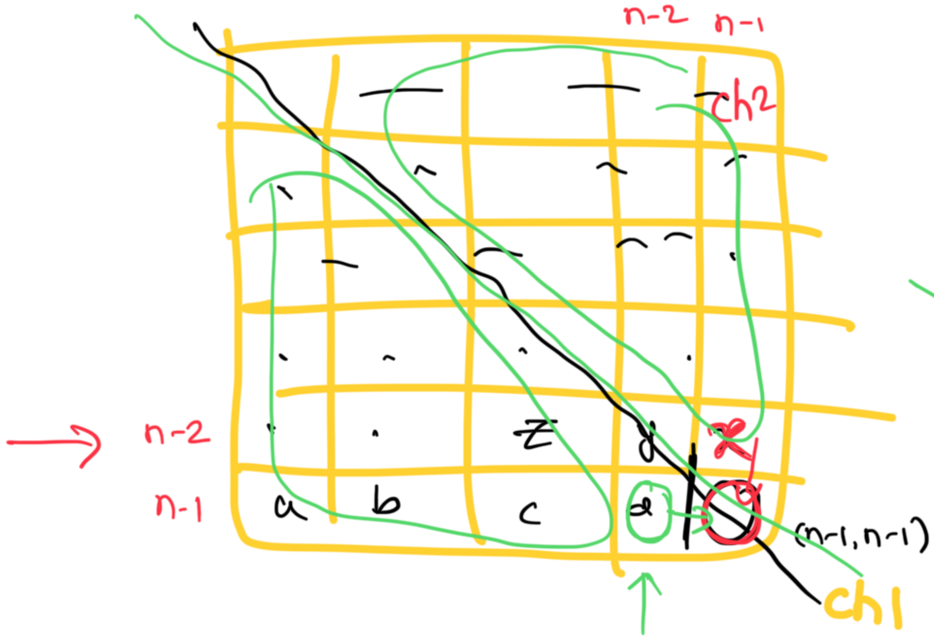
for(i = j+1 ; i<n ; i++) {

t[i][j] + max(t[i-1][j-1], t[i][j-1];
t[i+1][j-1]) ;

i+1 < n

}

}



child2 = t[n-2][n-1];   (0](n-1)

Child3 = t[n-1] [n-2] ;   (n-1] [0]

Child1 = diag sum;