# DP - Follow up

Question: Sell Wine

→ Sell these wines to make maximum profit.

→ We can either sell the left-most or right most wine in 1 year

→ Sell only one wine

| | | | | | |
|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 |
| A = | 2 | 3 | 5 | 1 | 4 |
| Year 2 | 2×2 | 6 | 10 | 2 | 8 |
| Year 3 | 6 | 9 | 15 | 3 | 12 |

5 Years

| | | | | | |
|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 |
| A = | 2 | 3 | 5 | 1 | 4 |

Year = 2
Year = 3
Year = 4
Year = 5

Strategy 1

profit = 2 + 6 + 12 + 4 + 25

= 24 + 25 = $\boxed{49}$

Strategy 2

A = ~~2~~ 3 5 ~~1~~ ~~4~~

Year = 2

Year = 3, 4, 5

profit = 2 + 8 + 3 + 12 + 25

= 25 + 25 $\boxed{50}$

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| A = | 2 | 3 | 5 | 1 | 4 |

Year = 1

Length = 5

Choices for 1st Year

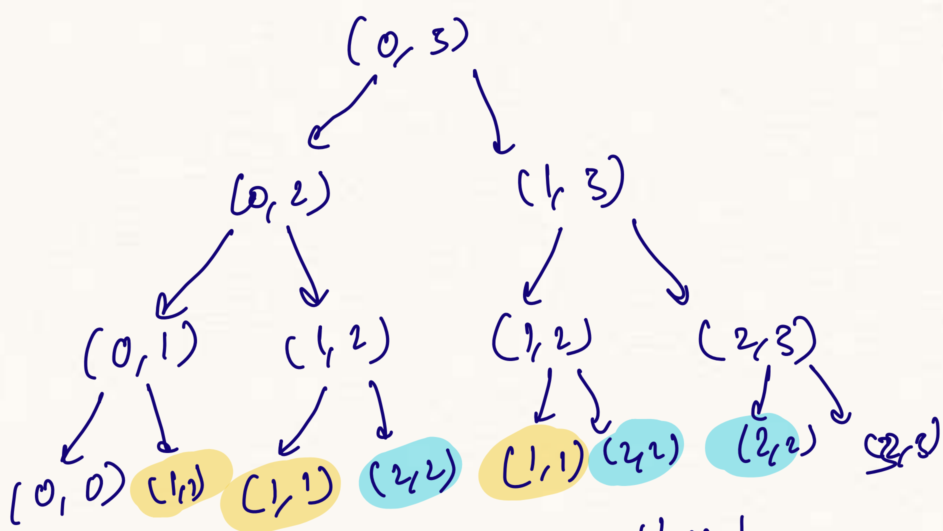
1) Sell the $0^{th}$ wine $\rightarrow$ $\boxed{2 + maxProfit(1, 4)}$ 2nd year

2) Sell the $4^{th}$ wine $\rightarrow$ 4 + maxProfit(0, 3) 1st year

$\boxed{maxProfit(2,2)}$

maxProfit( i, j) => Maximum profit I can make by selling the wines form A[i ... j]

$$maxProfit(i, j) = \max \begin{cases} A[i] * year + maxProfit(i+1, j)_{year+1} \\ A[j] * year + maxProfit(i, j-1)_{year+1} \end{cases}$$




Overlapping Subproblems !

Base Cases :

if( i = = j)

profit = A[i] x year

$\boxed{A[i] \times N}$

$N = 9$

$A = \underbrace{w_0 \quad w_1 \quad w_2}$ $\quad \overset{i=3}{w_3} \quad w_4 \quad \overset{j=5}{w_5}$ $\quad \underbrace{w_6 \quad w_7 \quad w_8}$

Total = $N$ $(=9)$

Remaining wines = $j-i+1$ $(=3)$

wines sold = $\underbrace{N-(j-i+1)}$ $\Rightarrow$ $(6+1)^{th}$ year

$= N-(j-i+1)+1$

$= N-j+i-\cancel{1}+\cancel{1}$

Year Number = $N-(j-i)$

```
int MaxProfit(int i, int j) {
    if(i==j) return N* A[i];

    if(dp[i][j] != -1) return dp[i][j];
    year = n - (j-i);
    // case 1
    P1 = year* A[i] + MaxProfit(i+1, j)

    // case 2
    P2 = year* A[j] + maxProfit(i, j-1)
    dp[i][j] = max(P1, P2);
}   return dp[i][j];
```

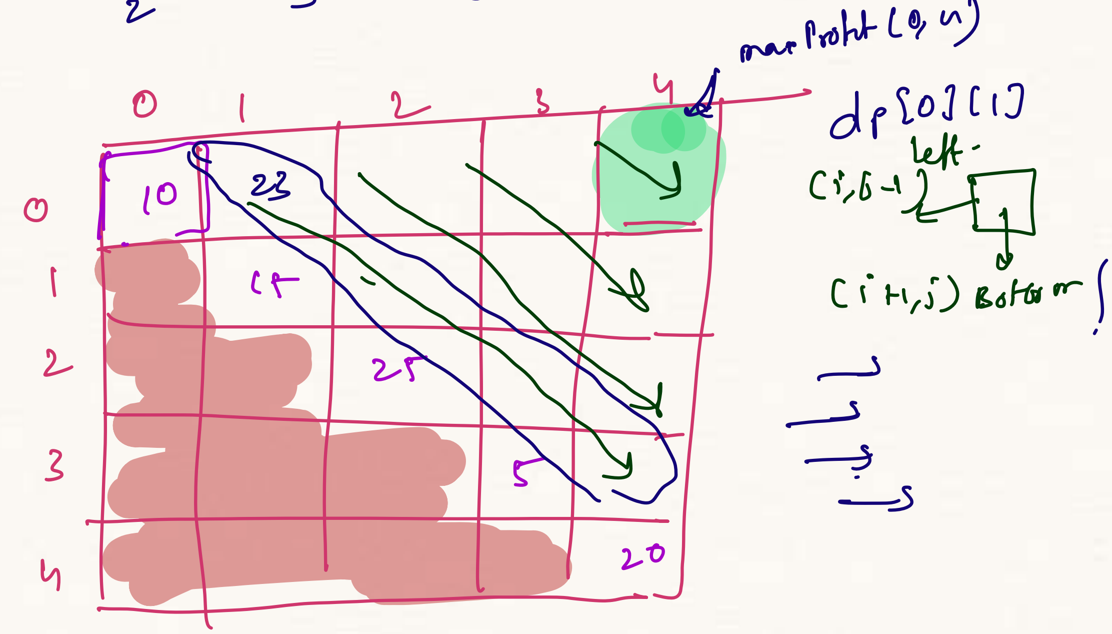
T.C: $O(\#\text{states} \times \text{T.C per state})$

$n^2 \times O(1) \Rightarrow \underline{O(n^2)}$

# Bottom up

$N = A.size()$

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| A = | 2 | 3 | 5 | 1 | 4 |




$year = N - 1 = 4$

$dp[0][1] = 4 \times A[0] + dp[1][1] = 8 + 15 = 23$

$\qquad\quad\;\; 4 \times A[1] + dp[0][0] = 12 + 10 = 22$

$Ans = dp[0][N-1]$

Question: No. of ways to partition a set into K subsets ( Bell Number)

Ex1 Input N = 3, K = 2 ✓

A = { 1, 2, 3 }

Ans → $\boxed{3}$

[ {1,2}, {3} ] , [ {1, 3}, {2} ] [ {2,3}, {1} ]

{ 1, 2, 3 } → { 1, 2, 3 }

{ 1, 2, 3 } → {1, 2},{3}
{1, 3} {2}
{2, 3}, {1}

Ex2 N = 3, K = 1

Ans = { 1, 2, 3 }

Ex3 $\boxed{N = 4, \quad K = 2}$

A = { 1, 2, 3, 4 }

[ [ {1, 2} {3, 4} ] [ {1, 3} {2, 4} ] [ {1, 4} {2, 3} ]
[ {1} {2, 3, 4} ] [ {2} {1, 3, 4} ] [ {3}, {1, 2, 4} ], [ {4}, {1, 2, 3} ]

Ans = 3 + 4 = $\boxed{7}$

Sol A = [ 1, 2, 3, 4 ] N = 4

$\boxed{N = 3, \quad K = 2}$

$\underbrace{[ \{1, 2\}, \{3\} ]}_{[2]}$ $\underbrace{[ \{1, 3\}, \{2\} ]}_{2}$ , $\underbrace{[ \{2, 3\}, \{1\} ]}_{[2]}$

2 + 2 + 2 = 2 × 3

{K}

No. of ways of partition 3 elements into 2 subset

Bigger Problem $N = 4$, $K = 2$ $\Rightarrow$

$A = 1, 2, 3, 4$

$N = 4$

Smaller Problems

{4}

{4}

Case 1: $N = 3$, $K = 2$

P1 $[\{1,2\}, \{3\}]$ P2 $[\{1,3\}, \{2\}]$, P3 $[\{2,3\}, \{1\}]$

4 4 4 4 4 4

ways(3, 2) = No. of ways of partitioning 3 numbers into 2 subsets.

$\boxed{\text{ways}(3,2) = 3}$ ← ways

P11 $[\{1,2,4\} \{3\}]$ P21 $[\{1,3,4\} \{2\}]$ P31 $[\{2,3,4\} \{1\}]$

P12 $[\{1,2\} \{3,4\}]$ P22 $[\{1,3\} \{2,4\}]$ P32 $[\{2,3\} \{1,4\}]$

$\Rightarrow$ $\boxed{\text{ways}(3, 2) \times K}$ $\Rightarrow$ $\text{ways}(N-1, K) \times K$

$\boxed{K = 2}$

{4}

Case 2: 4 comes as single in a set

$N = 4$, $K = 2$

$N = 1, 2, 3, | 4$

$\underbrace{1, 2, 3}_{n-1}$ → $k-1$ subsets

4 → 1 subset

$n-1$ elements into $K$ subset

$\boxed{\{4\}}$

$\Rightarrow$ ways$(n-1, k-1)$

Case1: Last Element goes into an existing subset

$ways(n, K) = ways(n-1, K) \times K$

Case2: Last elements goes into a new subset

$ways(n, K) = ways(n-1, K-1)$

$ways(n, K) \Rightarrow ways(n-1, K) * K + ways(n-1, K-1)$

Bell Numbers

$K = 2$
$K = 0$

Base Cases:

Way1

( N == 0 && K == 0 )
return 1

( K == 0 || N == 0 )
return 0;

N = 3

N > K

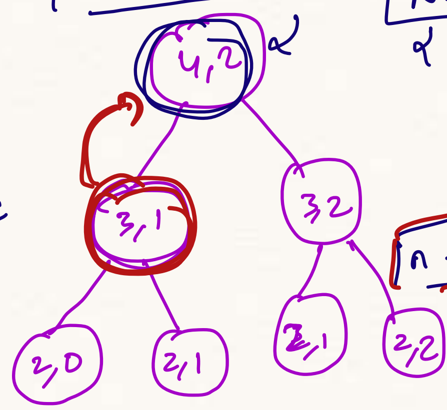
N > K

N = 4, K = 2

N > K

{1, 2, 3, 4, 5, 6}

n ≥ K




n < K.

Way2

{ if( N == K ) return 1;
if( K == 1 ) return 1;
if( N == 0

(K==0)! ever be reach if( N==1)

K = 1

$\boxed{(2,2)} \to (1,2)$

$\boxed{N_2 = K}$

Question : Div by 9

Given a number as a string

Find no. of subsequences of digits in N which form a number which is div by 9

$N =$ " $\underbrace{3\ 4\ 6\ 9\ 7}$ " → $10^5$ length

" $\underbrace{3\ 6}\ 9$ " , " 36 " " 9 "

$18 \% 9 = 0$ , $9\%9$ , $9\%$

Divisibility Rule for 9 : Sum of digits should be div by 9

[ (Sum of digits) % 9 == 0 ]

Div Rule for 3

Sum of digits % 3 == 0

$$\boxed{abcd = (a \times 10^3 + b \times 10^2 + c \times 10 + d) \% 3}$$

$(1234) \% 3 == 0$

1000

$$= (999a + a + 99b + b + 9c + c + d) \% 3$$

$$= \boxed{(a+b+c+d) \% 3}$$

$\boxed{7}$

9, 7, 5

$N = 4\ 3\ 4\ 6\ 9\ 7^{9}$

$N = 4\ 2\ 0\ 0\ 3\ 6^{9}$

{0036} {036} {36}

(sum of digits) % 9 = 0

Solution:

$N = \underbrace{3\ 4\ 6\ 9}\ 7$ (indices 0 1 2 3 4)

$(a+b)\%c = (a\%c + b\%c)\%c$

1) Include 7

$(7 + \text{rem\_sum})\%9 = 0$

$(7 + 4)\%9$

$(7 + \text{rem\_sum}\%9)\%9 = 0$

$\boxed{2} \to [0, 8]$

→ Choose elements such that the remainder when divided by 9 should be 2

Find no. of subsequences with remainder 2

2) Don't Include

Find no. of subsequences with remainder 0

F(rem, i) => No. of subsequences with the remainder 'rem' using the elements from A[0.....i]

**Choice 1:** Don't choose the ith element

$F(rem, i) = F(rem, i-1)$

**Choice 2:** Choose the ith element

| 0 | 1 | 2 | 3 |
|---|---|---|---|
| 3 | 4 | 6 | 7 |

$(10-9)$

$F(3, 3)$

rem = 3 (sum % 9 = 0)

7 +

Let 'x' be the remainder to look in the remaining (i-1) elements

$(x+7)\%9 = 3$

$(x + A[i])$

[0 ... i-1]

A = 

| 0 | 1 | 2 | 3 |
|---|---|---|---|
| 3 | 4 | 6 | 7 |

rem = 3

0, 1, 2 ... 8

$F(rem = 3, i = 3)$

$(7 + x)\%9 =$

Let x be the remainder of the remaining element

$((x+7)\%9 = 3$

$\boxed{x = 5}$

$x = (rem - A[i] + 9)\%9$

$\rightarrow (rem - A[i])\%9$

$(3 - 7)\%9$

$(x + A[i])\%9 = rem$

$x = (rem - A[i])\%9$

$F(rem, i) = F(rem, i-1) + F((rem - A[i] + 9)\%9, i-1);$

$\boxed{A[i]\ \%\ 9}$

$A[i] = \{0 \dots 9\}$

rem=8

$[0 \dots N-1]$

$[0 \dots 5]$

Base Cases :

rem = 3

rem = 0

i = 5

if( i < 0 && rem == 0)

0 or $\boxed{1}$

70% 30%

0 1

$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{array}$

$\boxed{i = 0}$

[1]

$\boxed{i < 0}$

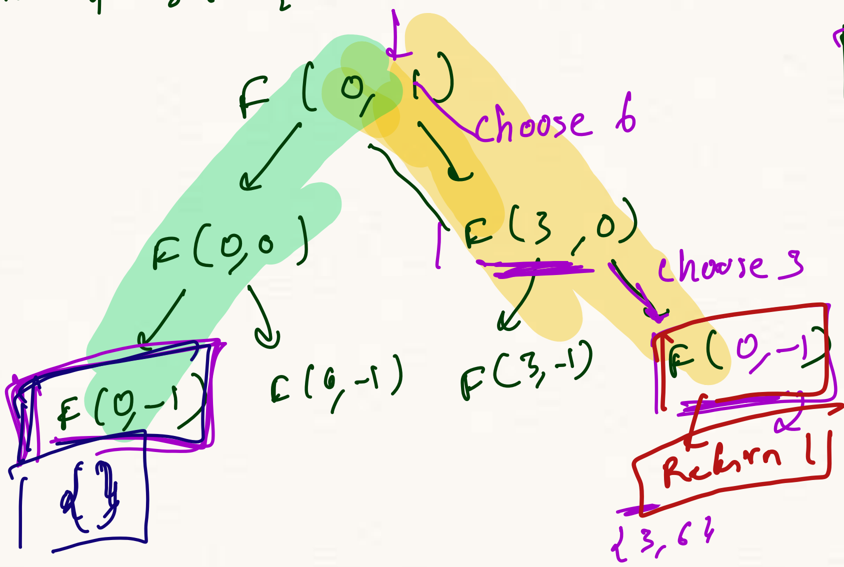
$(3 - 3 + 9)\%9 =$

0 in $A[0 \dots 1]$

$N = \begin{array}{cc} 0 & 1 \\ 3 & 6 \end{array}$

No. of subset with remainder

$\boxed{(rem - A[i] + 9)\%9}$




$(0 - 4 + 9)\%9$

$5\%9 = 5$

$(0 - 6 + 9)\%9$

$3\%9$

$(3 - 3 + 9)\%9$

$(0 - 3 + 9)\%9$

6

$\boxed{ans = F(0, N-1) - 1}$

3 4 6 7

7 +

sum = A[i] + [i-1.]

thursday

Question: Dungeon Princess

health

health = 10 6 2

health ≥ 1

$$
A = \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & -2 & -3 & 3 \\ 1 & -5 & -10 & 1 \\ 2 & 10 & 30 & -5 \end{array}
$$

Princess

Health = 3

Health = 1

Halth = 5

2+3 = 5

5+1 = 6

6 − 5 = $\boxed{1}$

Health = 7

Health = 6

= 4

= 1 + 3

= 4

= 5

5 − 5 = 0

health > 0

1 − 2 = −1

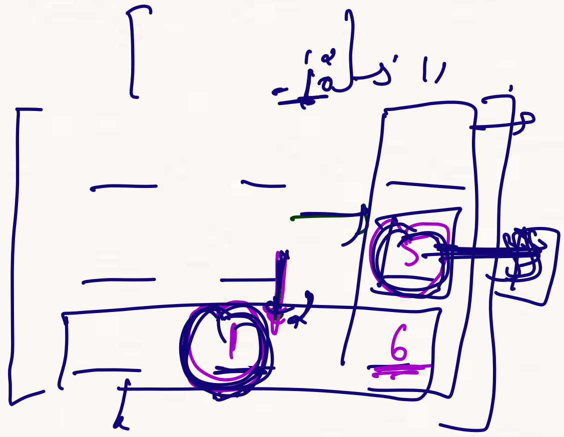
health = 2

health = 3

$\boxed{6}$

$$A = \begin{bmatrix} -2 & -3 & 3 \\ -5 & \boxed{-10} & 1 \\ 10 & \boxed{30} & \boxed{-5} \end{bmatrix}$$

5   6   -24



health ≥ −24
health ≥ 1

$|-24| + 30 = 6$

minHealth(i, j) = the minimum health required to enter the cell (i, j)

Let $x$ be the health before entering cell (i, j)

$x + A[i][j] \geq \min(\ ,\ ) - A[i][j]$

$x - 10 \geq \min(1, 5)$

$x \geq \min(1, 5) + 10$

$x \geq \min(\text{minHealth}(i+1, j), \text{minHealth}(i, j+1)) - A[i][j]$

$x \geq \min(\text{minHealth}(i+1, j), \text{minHealth}(i, j+1)) + A[i][j]$

$x \geq 1$

$x \geq \max(1, \ );$

$\boxed{\begin{matrix} x \geq a \\ x \geq b \end{matrix}}$

$x \geq \max(a, b)$

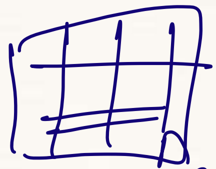
$x \geq 2, \quad x \geq 3$

$x \geq \max(2, 3)$

dp[i][j] = Min Health required to enter the cell (i, j) so that we save the princess

$dp[i][j] = max(1, min(dp[i][j+1], dp[i+1][j]) - A[i][j])$

Base Case



n x m

Ans = [dp[0][0]]

dp[n-1][m-1]

Base Case

if( A[n-1][m-1] > 0 )

dp[n-1][m-1] = 1

else

dp[n-1][m-1] = abs( A[n-1][m-1] ) + 1

if( i ≥ n || j ≥ m)

return INT_MAX;

sum = 29

remainder

Ans

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| N = | 3 | 4 | 6 | 9 | 7 |

22

$$F(sum, i) = \begin{bmatrix} F(sum - A[i], i-1) \\ F(sum, i-1); \end{bmatrix}$$

≥ 0

(29, 4)

(22, 3)    (29, 3)

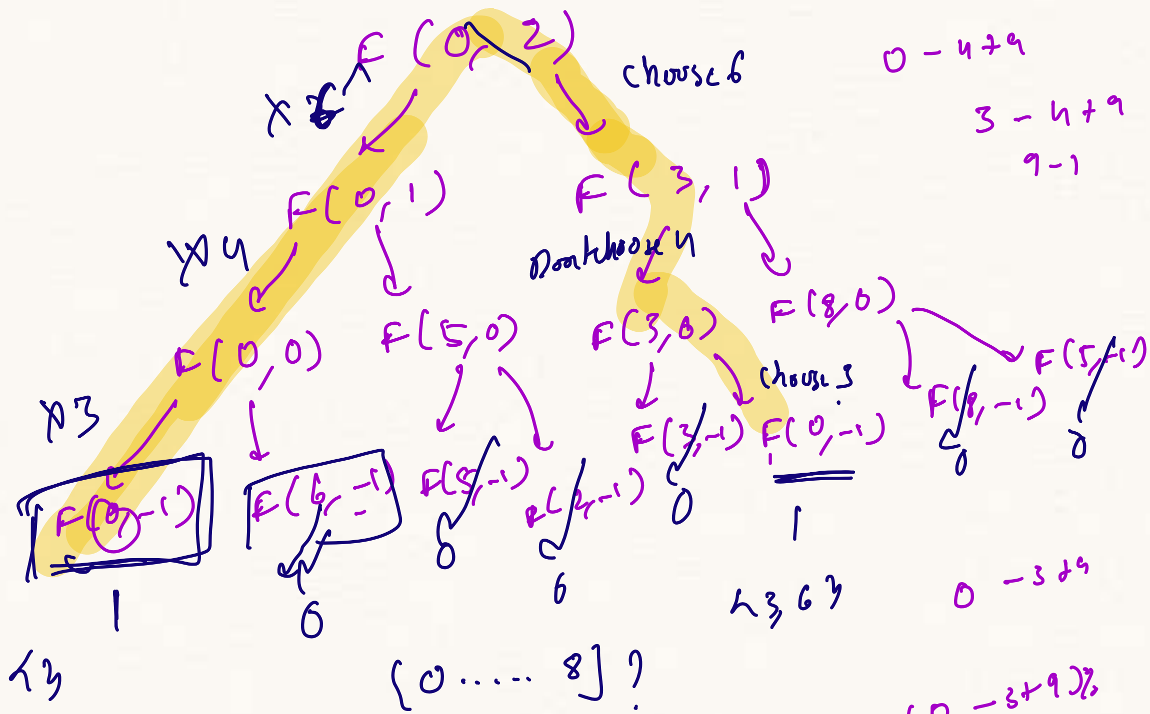
(13, 2)    (22, 2)

$$A = \begin{array}{ccc} 0 & 1 & 2 \\ 3 & 4 & 6 \end{array}$$

rem = 0

(rem - A[i] + 9) % 9

(0 - 6 + 9)




0 - 4 + 9

3 - 4 + 9

9 - 1

0 - 3 + 9

(0 - 3 + 9) %

5 = 3 + 9

2 : 9

Subtract 1

rows =

dp[rem] [N]

{0 .... 8}

Array length

$$N = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array}$$

$$A = \begin{array}{ccccc} 3 & 4 & 6 & 9 & 7 \end{array}$$

sum = 0

7 + $\boxed{7 \% 9}$

Non subsequence divisible by 9 such that current subseq has a sum 9 zero

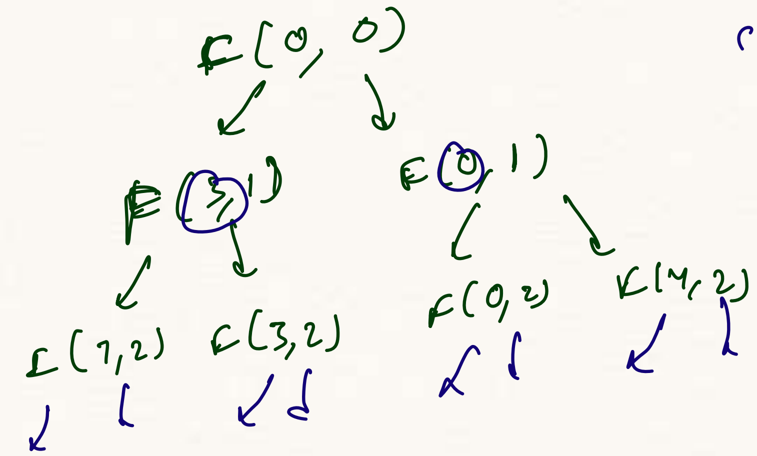
$\boxed{F(sum, i)}$ = F( (sum + A[i]) % 9 , i + 1 )

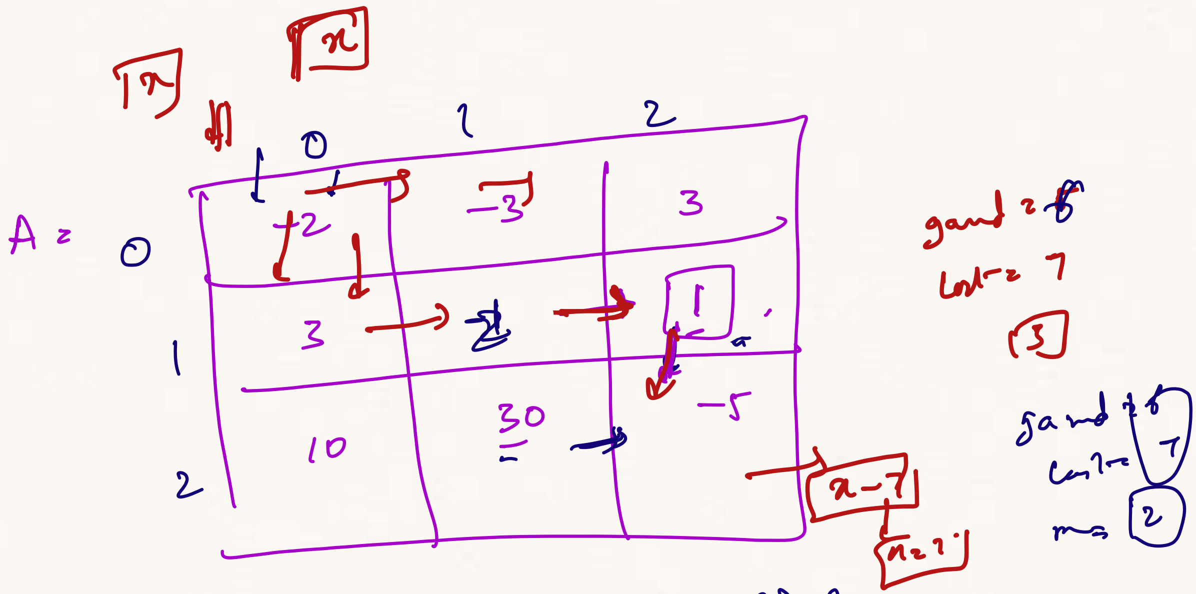
remainder

F(sum, i + 1)

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 3 & 4 & 6 & 9 & 7 \end{array}$$

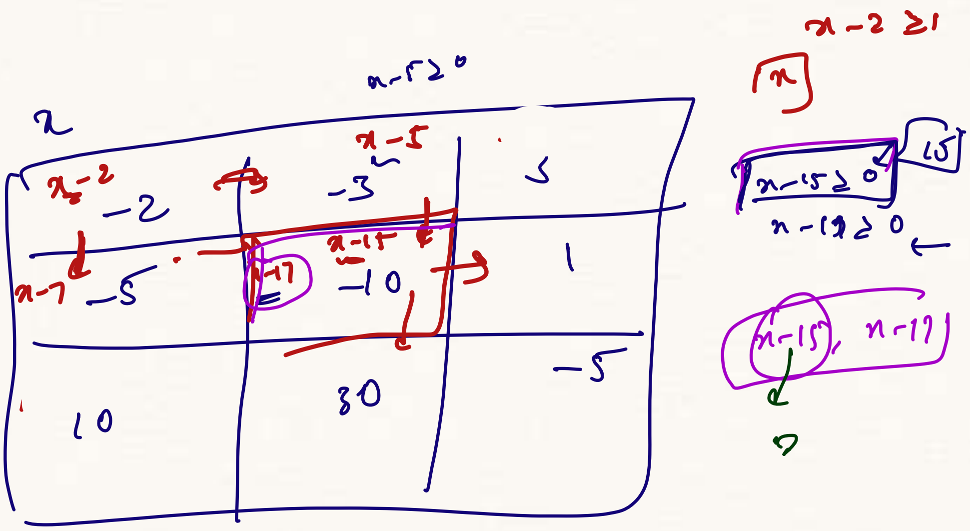
[0 .... 8]




$dp[i][j] = dp[i+1][j] + dp[i][j+1]$

dp[



$$A = \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & -2 & -3 & 3 \\ 1 & -5 & -10 & 1 \\ 2 & 10 & 30 & -5 \end{array}$$

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 3 & 6 & 6 \\ 1 & 8 & 16 & 8 \\ 2 & 8 & 8 & 11 \end{array}$$

dp[i][j] = min( dp[i-1][j], d[i][j-1] − A[i][j] )

Min Health to reach (i,j) from (0,0)

$\leq 0$

**Sell Wine : Bottom-Up**

```
dp[N][N] = {0};
for(int i = 0; i < n; i++)
      dp[i][i] = price[i] * n;

for(int len = 2; len <= n; len++){
      for(int i = 0; i <= n - len; i++){
            int j = i + len - 1;
            year = n - len + 1;
            int x = price[i] * year + dp[i+1][j];
            int y = price[j] * year + dp[i][j-1];
            dp[i][j] = max(x, y);
      }
}
return dp[0][n-1]
```

**Div by 9**

```
int num(int rem, int i, string &A){
    if(i == -1){
        if (rem == 0) return 1;
        return 0;
    }

    if(dp[rem][i] != -1) return dp[rem][i];
    dp[rem][i] = (num((rem - (A[i] - '0') + 9) % 9, i - 1, A) +
num(rem, i - 1, A))%mod;

    return dp[rem][i];
}
```

**Sell Wine : Top-Down**

```
int maxProfitUtil(vector<int> price, int i, int j, int n) {
      if(i == j) return n * price[i];

      if (dp[i][j] != -1)
        return dp[i][j];

    int year = n - (j - i);
    int x = price[i] * year +
            maxProfitUtil(price, i + 1, j, n);

    int y = price[j] * year +
            maxProfitUtil(price, i, j - 1, n);

      dp[i][j] = max(x, y);

    return dp[i][j];
}
```