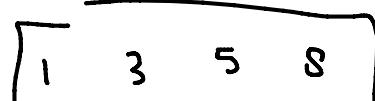


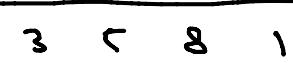
Dynamic ProgrammingBurst Balloons

$$[3, *, 5, 8] \rightarrow$$



$$\downarrow$$

$$[3, 5, 8]$$



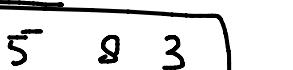
$$\downarrow$$

$$! [3, 8]$$



$$\downarrow$$

$$! [8]$$



$$\begin{aligned} 3 \times 1 \times 5 &= 15 \\ 3 \times 5 \times 8 &= 120 \\ 3 \times 8 \times 1 &= 24 \\ 3 \times 1 \times 1 &= 3 \end{aligned}$$

$$\begin{array}{r} 120 \\ 24 \\ 24 \\ \hline 162 \end{array}$$

$$[3, *, 5, 8]$$

$$! \tau 38$$

$$3 \times 5 \times 1 = 15$$

$$3 \times 1 \times 8 = 120$$

$$[3, 5, 1, 8]$$

$$1 \times 3 \times 5$$

$$1 \times 3 \times 8 = 24$$

$$(\times 3, 8)$$

Subproblem
Independent

$$\boxed{\quad} : \boxed{\quad} =$$

$$\begin{array}{r} 120 \\ 24 \\ 24 \\ \hline 162 \end{array}$$

$$[8] = 1 \times 8 \times 1 = 8$$

$$!$$

$$[3, 1, 5, 8]$$

$$[1, 5, 3, 8]$$

$$162$$

$$\begin{array}{r} 1 \times 3 \times 1 \\ \hline 3 \end{array}$$

$$[3, 1, 5, 8]$$

$$X =$$

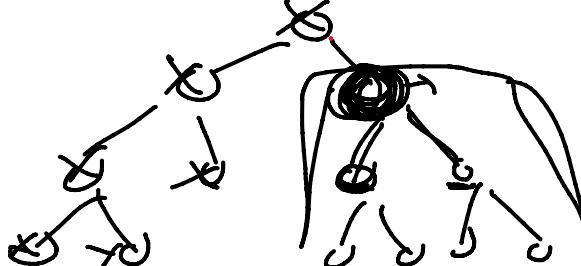
$$[1, 5, 6]$$

$$1538$$

$$162$$

$$[3, 1, 5, 8]$$

$$|$$



$$\begin{array}{r} 6 \\ 6, 6, 6 \end{array}$$

$\lceil 3 \ 1 \ 5 \ (8) \rceil$

$\lfloor 3 \ 5 \ 8 \rfloor$

$\lceil 3 \ 8 \rfloor$

$\lceil 8 \rfloor$

$\lceil 1 \ 5 \rfloor$

$\lceil \downarrow \downarrow \rfloor$



$1 \swarrow \searrow \downarrow \downarrow$

last builson Phadeep

$\lceil 8 \rfloor$

$\lceil 3 \ 8 \rfloor$

$1 \times 8 \times 1$

$(x \ 3 \times 8)$

$3 \times 5 \times 1$

$\lceil 5 \rfloor$

$\lceil 3 \ 5 \ 8 \rfloor$

int f (int i, int j, vector<int>&a)

{ if (i > j)

return 0;

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

int maxi = INT-MIN;

for (int i=0; i < j; i++)

{ int temp = $a[i+1] + a[i+2] + a[j+1]$ +

$f(i, \underline{i+1}, a) + f(\underline{i+1}, j, a);$

maxi = max(maxi, temp);

dp[i][j] =
order, max x

[bu]

[b3 b4]

.

$b_1 \ b_2 \ b_3$

$b_4 \ b_5$

b_6

b_7

3 last element

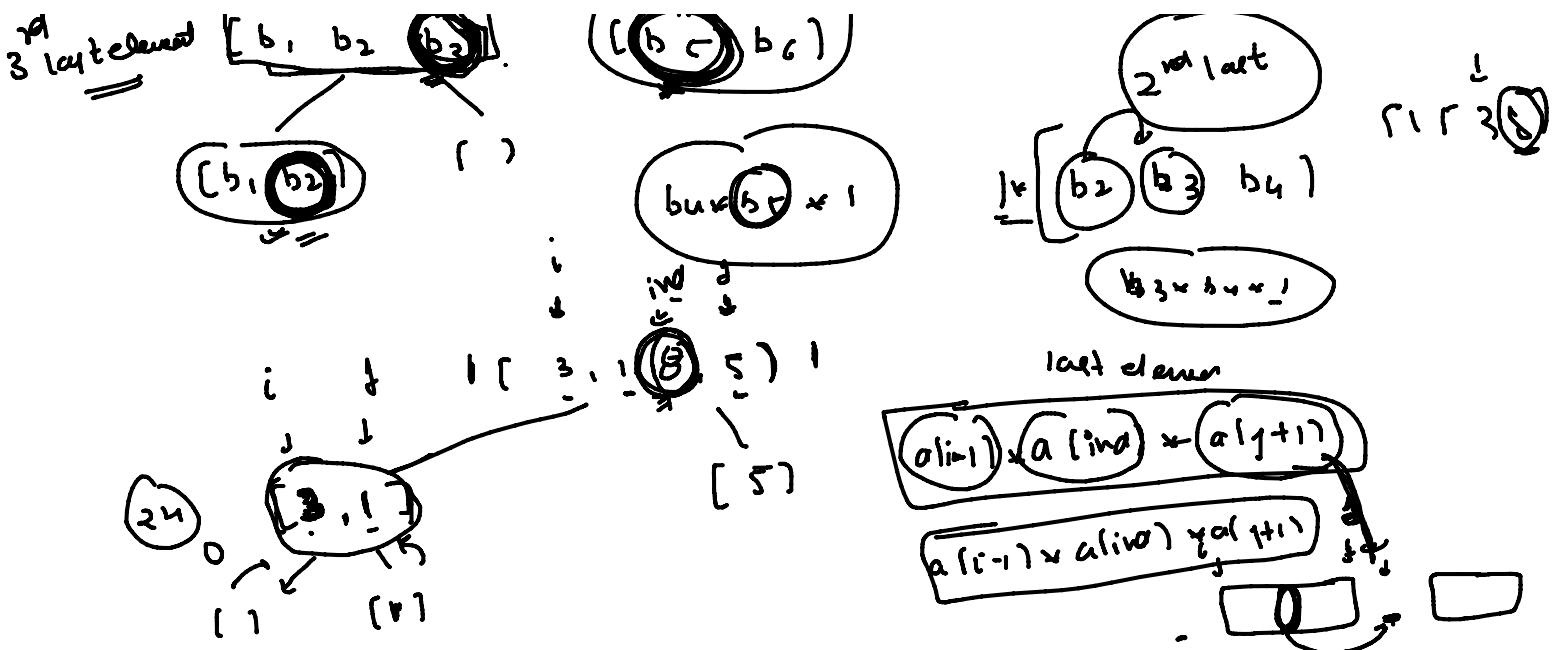
$\lceil b_1 \ b_2 \ b_3 \rfloor$

$\lceil b_4 \ b_5 \ b_6 \rfloor$

$1 \times b_3 \times b_4$

2nd last

1



For ($i = n - 2$; $i \geq 1$; $i--$)

↳ For ($j = i$; $j \leq n - 2$; $j++$)

↳ ↳

$\text{int max} = \text{INT_MIN};$

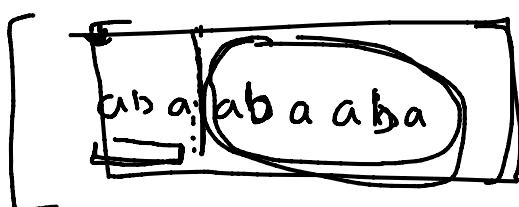
For ($\text{int mid} = i$; $\text{mid} \leq 1$; $\text{mid} + 1$)

↳ $\text{int cost} = a[i+1] * a(mid) + a(1+1) + \text{up}[i][\text{mid}-1] +$
 $a[i+1][1];$

↳ $\text{max} = \text{max}(\text{max}, \text{cost});$

$\text{up}[i][j] = \text{max};$

Palindrome Partitioning II



Front Partition P

$f(i)$

$O - n$

int f (int i, int n, string str)

1 if (i == n) return 0; \rightarrow if (dp[i] == -1) return dp[i]

int minCost = INT_MAX;

For (j = i ; j < n ; j++)

1 if (isPalindrome (i, j, str))

2 int cost = 1 + f (j + 1, n, str);

minCost = min (minCost, cost);

1 \downarrow

$dp[i] =$

1 return minCost;

①

ab a | ab a || ab a | : →
ab a ab a

$dp[i]$

$i \rightarrow n-1$

$dp[n] = emptyString = 0$

i
 $0 - n$

$(n-1) \rightarrow 0$

vector<int> dp(n);

for (i = 0 ; i < n ; i++)

vector<vector<int>> dp;

dp[n] = 0;

For (i=n-1; i>=0; i--) int mincost = INT_MAX;

 For (j=i; j<=n; j++)

 if (ispaired(str, i, j))

 int cost = 1 + dp[j+1];

 mincost = min(mincost, cost);

 dp[i] = mincost;

return dp[0];

Sieve subtraction without ans

0 1 1 1 1

1 1 1 1 1

0 1 1 1 1

vector<vector<int>> dp(r, vector<int>(n, 0));

for (j=0; j<r; j++) dp[0][j] = arr[0][j];

for (i=0; i<n; i++) dp[i][0] = arr[i][0];

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

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

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

if ($\text{arr}[i] \geq \text{arr}[j] == 0$) $\text{dp}[i][j] = 0$;

else

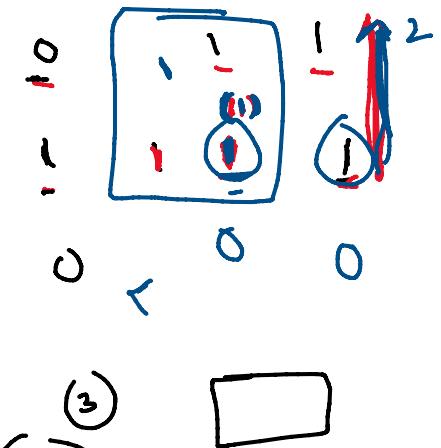
$\text{dp}[i][j] = 1 + \min(\text{dp}[i-1][j], \min(\text{dp}[i-1][j-1],$
 $\text{dp}[i][j-1]);$

2

n

=>

0	1	1	1
1	1	2	2
0	0	0	6



$\text{dp}[i][j] = \text{length of longest suffix subsequence } i:j$

Find max of

dp vector