DP-4

Agenda

- → Length of LIS → $O(n^2)$ → DP 2
- → Russian Doll Envelopes
- → No. of B.S.T
- → Matrix Multiplications

Question: Length of LIS

Discussed: $O(n^2)$ solution using DP

$N = 10^5$ → $(10^5)^2 = 10^{10}$

Ex: 10 1 5 7 3 9 4 13 11 12

LIS: 1 5 7 9 11 12 → $\boxed{L = 6}$

temp[1], temp[2] . . .

temp Array

→ 1 - indexed

→ $\boxed{\text{temp[i]}}$ Stores the smallest number that ends a increasing subsequence of length i

A = 5 2 4 6

The subseq of length 2 is {5,6} {2,4} {4,6}

6 4 6

temp[2] = 4

[2, 4, 6]

temp[3] = 6

temp[1] = 2

temp[4] = INF

{5} {2} {4} {6}

{1, 5, 7, 9, 11, 12}

{1, 3, 4, 11, 12}

A = 10 1 5 7 3 9 4 13 11 12

2

{1 }

temp 1 3 4 9 11 12 _ _

1 2 3 4 5 6 7 8

len = 5 6

{10} {1}

{1, 5, 7}

{1} {1, 5} {1, 5, 7} {1, 3, 7, 9} {1, 3, 7, 9, 13}

{1, 3}

temp[2] = 5

Binary Search

upper bound

ceiling

temp[3] = 5

{1, 3, 5}

4

{1} {10}

1 3 5

1 3 4

4

→ we can only find length that LIS

→ we cannot find what

A = 10 1 5 7 3 9

upper bound

{1, 5}

temp = 0 1 5 7 _ _ _

1 2 3 4 5

len = 3

L=1 {1} {1, 5} {1, 5, 7}

L=2

{1, 3}

{5}

tail[i]

Largest Num < less

T.C: O(n log n)

O(log n)

Question: Russian Doll Envelopes

N envelopes each with a height & width

Env i (hi, wi) can be put into Env j (hj, wj) only if

$h_i < h_j$ && $w_i < w_j$

E1 2 x 3 ✗ E2 3 x 3

E1	E2	E3	E4
6 x 4	5 x 4	6 x 7	2 x 3

E1 E2 E4

E4 2 x 3 → E2 5 x 4 → E3 6 x 7

3

heights: (2 < 5 < 6)

widths: (3, 4, 7)

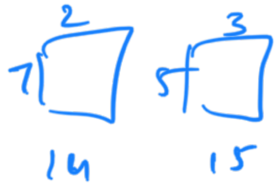
h = 5 3 6 2

7 2 6

w =) ,

Sort h = 2 3 5 6

w = 6 7 1 2



E1 <- E2

E2 -> E1

j = i-1

-> we want to select envelopes such that height & width are ↑

h w

in = (6, 4) (5, 4) (6, 7) (2, 3)

(3, 4, 7)

Sort(A) = (2, 3) (5, 4) ~~(6, 4)~~ (6, 7)

-> choose Max No of envelops such that widths are increasing

3 4 4 7

{3, 4, 7}

Apply LIS on the width

option1

(2, 3) ✓
(5, 4) ✓
(6, 7) ✓

option2

(2, 3) ✓
(6, 4) ✓
(6, 7) ✓

$\boxed{LIS}$

T.C : O(nlogn) + O(nlogn)

BS approach

→ ↓ Sort based on height.

(2,3) (3,4) ~~(6,5)~~ (6,7) (7,7)

$\boxed{3\ 4\ 5\ 7}$

LIS based on width:

{3, 4, 5, 7}

$\boxed{3, 4, 5, 7}$ → $\boxed{4}$

(2,3) (3,4) ~~(6,6)~~ (6,7) (7,7)

$\boxed{3, 4, 7, 7}$ → 3, 4, 7

solution: while finding LIS on widths, we can check the heights as well.

$\boxed{w[i] > w[j] \ \&\& \ h[i] > h[j]}$

Question: Russian Doll Env with Rotation

You can rotate env

	E1	E2	E3
H	1	9	5
W	3	4	10

(E1 → E2) (E1 → E3)

No Rotations :

Rotation:

$(E_1 \to E_2 \to E_3)$

H	1	4	5
W	3	9	10

A =	H	1	3	9	4	5	10
	W	3	1	4	9	10	5

(1, 3)

E_1 R(E)

(2, 3) (3, 2)

h < w

w < h

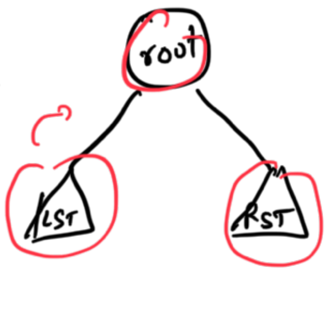
E_1 E_2

$(7, 5) \to (5, 7)$

h < w or w > h

Question: No. of B.S.Ts

Given N, find no. of unique B.S.Ts that can be created using numbers [1, 2 .... N]



N = 1

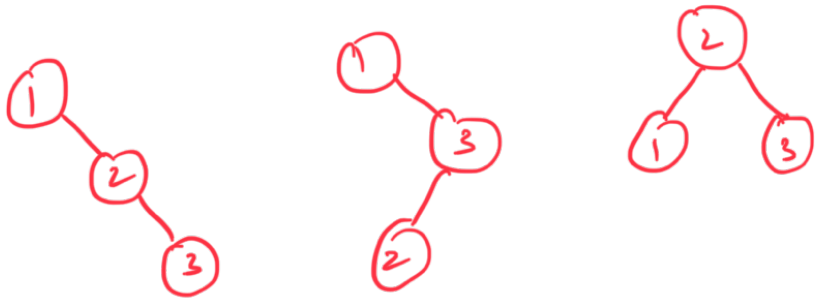
→ 1

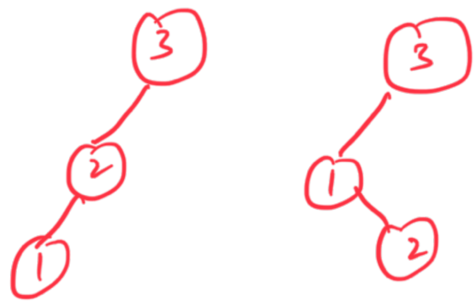
N = 2

→ 2

N = 3

[1, 2, 3]





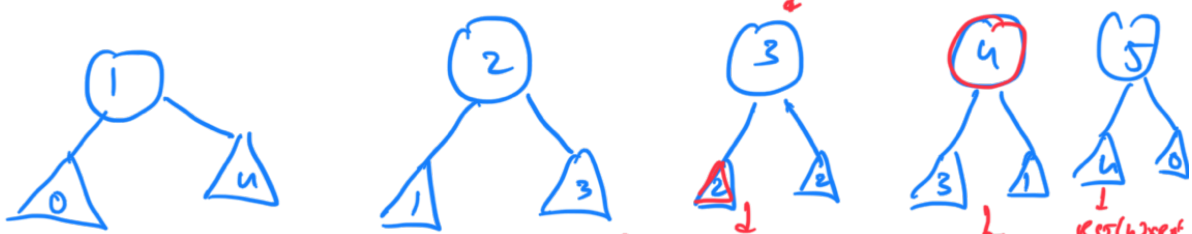
→ 5 BSTs

A = [1, 2, 3]   A [11, 12, 13]

BST(i) → Denotes no. of B.STs I can create using N element

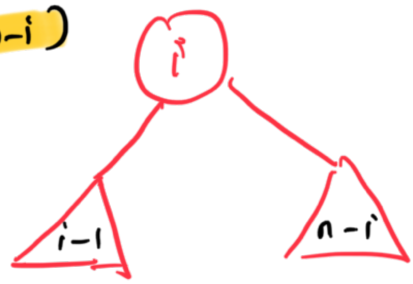
N = [1, 2, 3, 4, 5]

→



BST(0) * BST(4)   BST(1) * BST(3)   BST(2) * BST(2)   BST(3) * BST(1)   BST(4) * BST(0)

$$BST(n) = \sum_{i=1}^{n} BST(i-1) * BST(n-i)$$



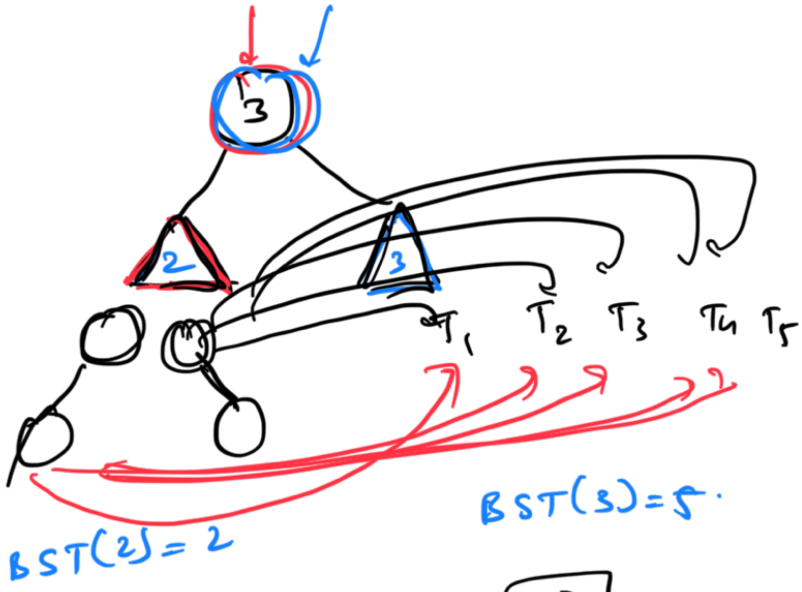
$(i-1) + 1 + x = n$

LST   root   RST

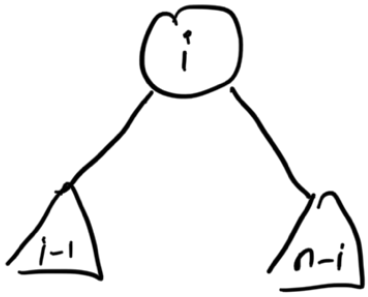
$x = n - i$

Let us find no. of BST's with 'i' as the root.

N = 6, i = 3



$2 \times 5 = \boxed{10}$



$BSTs(i-1) \times BSTs(n-i)$

$$BST(n) = \sum_{i=1}^{n} BST(i-1) \times BST(n-i)$$

Overlapping subproblems? YES

if(N == 0) return 1;

if(N == 1) return 1

T.C: O(#State × T.C Per State)

N

$O(N)$

T.C : $O(N^2)$

S.C : $O(N)$

$$\boxed{C(n) = \sum_{i=1}^{n} C(i-1) \times C(n-i)} \rightarrow O(N^2)$$

Catalan Numbers

$$C(n) = \frac{{}^{2n}C_n}{n+1} \rightarrow O(n)$$

$$^{n}C_r = \frac{n!}{r!\,(n-r)!}$$

$$\frac{\dfrac{2n!}{n!\,.\,n!}}{n+1}$$

$2n! \rightarrow O(2n)$

$n!\,.\,n! \rightarrow O(n)$

T.C : $(N!) = O(N)$

$1 \times 2 \times 3 \ldots\ldots n$

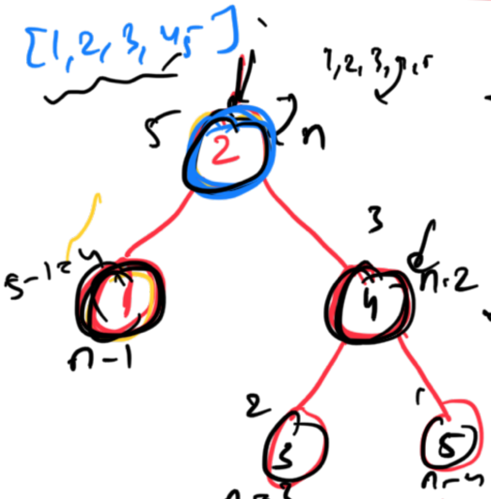
$O(n) \rightarrow 2n!,\ n!$

$$\boxed{\frac{2n!}{n!\,n! \times n+1}}$$

$N = 5$

[1, 2, 3, 4, 5]

[1, 2, 3, 4, 5]



General form

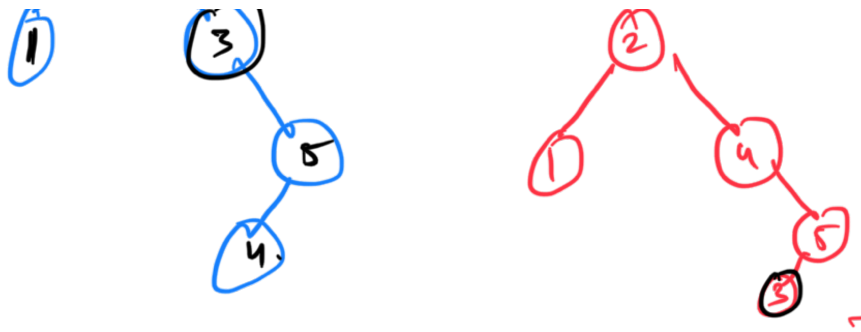
$N = 5$

$\boxed{n!}$

$n \times (n-1) \times (n-2) \ldots 1 = \boxed{n!}$

Structure of the tree is fixed

$N = [1, 2, 3, 4, 5]$



Question: No of Binary Trees using N numbers.

Binary Trees = $\boxed{\text{\# B.STs}}$ × N!

Catalan Num × N!

Every Tree Structure → 1 B.S.T

$\boxed{\text{\# B.S.T} \times n!}$

N = 5

Question: Matrix Chain Multiplication

$[\ \ ]_{a \times b} \quad [\ \ ]_{c \times d} \quad [\ \ ]_{m \times n}$

$m \times n \to a, d$

$b == c$

cols(M1) = Row(M2)

$m = a$

$n = d$

$M1 \times M2 \neq M2 \times M1$

i)

$$\underset{M1}{\begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \\ c_1 & c_2 \end{bmatrix}}_{3 \times 2} \times \underset{M2}{\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{bmatrix}}_{2 \times 4} = \begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix}_{3 \times 4}$$

$a_1 . x_1 + a_2 . y_1$

$a_1 x_2 + a_2 y_2$

M1 $(a \times b)$ M2 $(b \times c)$

cells =>

$a . c$

cost for 1 cell = b

Total cost = $a \times \underline{b} \times c$

M2 M1 M3 M4

M1 $(5, 4)$ M2 $(4, 6)$ M3 $(6, 2)$ M4 $(2, 7)$

M1 . M2 . M3 . M4

2

((M_1 . M_2) M_3) . M_4

Cost

5.4.6 = 120

(5, 6)

(5, 2)

(5, 7)

Cost	Cost
120	120
+	84
60	210
+	
70	
$\boxed{250}$	414

((M1 . M2) (M3 . M4))

(5, 6)

(6, 7)

(5, 7)

(M1 . (M2 . (M3 . M4)))

		M1	M2	M3	M4
d =	5	4	6	2	7
	0	1	2	3	4

dim(M1) = d[0] x d[1] — (5 x 4)

dim(M2) = d[1] x d[2] → 4 x 6

dim(M3) = d[2] x d[3] → 6 x 2

$\dim(M_i) = \underbrace{d[i-1]}_{\text{rows}} \times \underbrace{d[i]}_{\text{colm}}$

M1 . M2 . M3

(M1 . M2) . M3 — c_1

M1 (M2 . M3) — c_2

N = 4

d

M1 M2 M3 M4

M1 M2 M3 M4

$M_1 \mid M_2 \quad M_3 \quad M_4$

1   3

C_1

2   2

C_2

3   1

C_3

min $[C_1, C_2, C_3]$

cost(i, j) → The minimum cost for multiplying matrices $M_i \ldots M_j$

k = 2

M_1	M_2	M_3	M_4
5×4	4×6	6×2	2×7

$[(M_1 . M_2) \times (M_3 . M_4)]$

(5, 6)   (6, 7)

cost(1, 4)

Case 1   Partition at k = 2

→ cost(1, 2) + cost(3, 4) + 5 × 6 × 7

$r_i \quad c_k \times c_j$

$d[i-1] . d[k] . d[j]$

N = 5

k = 1   k = 2   k = 3   k = 4

M1	M2	M3	M4	M5
5×6	6×2	2×3	3×4	4×2

cost = cost(1, 3) + cost(4, 5) + 5 × 3 × 2

cost(1, 3) + cost(4, 5) + 30

(i, j)

partition at 'k'

Cost for a partition

cost(i, k) + cost(k+1, j) + d[i-1] × d[k] × d[j]

1, N   k = 1 to N-1

```
dp[ ][ ]
int  minCost( i, j ) {
        if( i == j ) return 0;
        dp[i][j] != -1  return dp[i][j]
        ans = INT_MAX;
        for( k = 1 ;  k ≤ j-1 ;  k++ ) {
  O(1)       ans = min( ans,  minCost( i, k )
                          + minCost( k+1, j )
                          + d[i-1] × d[k] × d[j] );
        } dp[i][j] = ans;
        return ans;
}
```

minCost(1, 4);
d.size() - 1;

	M1	M2	M3	M4
d = 5	4	6	2	7

$\boxed{i^{th} : d[i-1] \times d[i]}$

T.C: O(#state × T.C per state)

$N^2 \times N$

: $O(N^3)$

S.C: $O(N^2)$

	M1	M2	M3	M4
d = 5	4	6	2	7

Bottom-Up

	1	2	3	4
1	0	$\boxed{120}$		
2		0		
3			0	
4				0

$dp[i][j]$ = min cost for multiply matrices from $M_i \dots M_j$

$(M_1 M_2)$ $(M_2 M_3)$ $(M_3 M_4)$

$(M_1\ M_2\ M_3\ M_4)$

$\downarrow$

$M_1 M_2 M_3$ $\quad$ $M_2 M_3 M_1$

$$\begin{array}{lccccc} & & M_1 & M_2 & M_3 & M_4 \\ d = & 5 & 4 & 6 & 2 & 7 \\ & 0 & 1 & 2 & 3 & 4 \end{array}$$

$dp[1][2] = \underset{0}{dp[1][1]} + \underset{0}{dp[2][2]} + \underset{120}{5 \times 4 \times 6}$

$i = 1, j = 2$

$k = [1]$

$dp[2][3] =$

(i, j)

$(k = i ; k \le j-1 ; k++)\{$

$dp[i][j] = \min(dp[i][j],$
$dp[i][k] + dp[k+1][j] +$
$d[i-1] \times d[k] \times d[j]$

$\}$

Question: Count of Valid Parenthesis using

N pairs

N=1 ( ) → 1

N=2 ( ) ( ) (( )) → 2

N=3 ((( ))) ( ) ( ) ( ) (( ) ( )) ( ) (( )) (( )) ( ) → 5 →

$O(2^N)$

BST(1) = 1
BST(2) = 2
BST(3) = 5

→ $\boxed{C_n} = \frac{{}^{2n}C_n}{n+1}$

length = 2n

N pairs

( ) _) _) _)
1 2 3 4 5 6 7 8

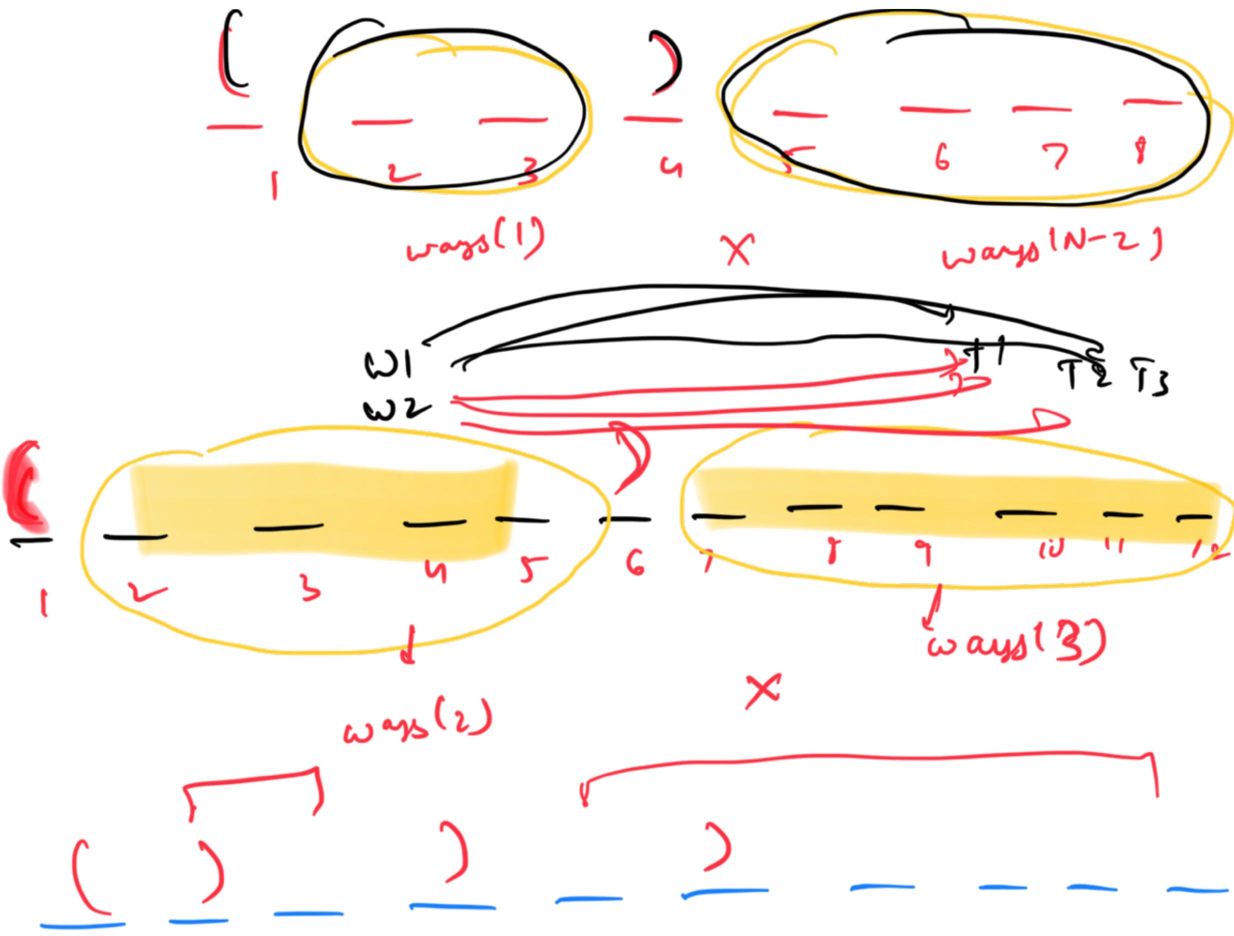
( )

ways(N) = No. of balanced parenthesis we can create using N pairs

( )
1 2 3 4 5 6 7 8

(N−1) pair

ways(N−1)



$$\text{ways}(N) = \text{ways}(0) \times \text{ways}(N-1) + \text{ways}(1) \times \text{ways}(N-2) + \text{ways}(2) \times \text{ways}(N-3) + \cdots$$

$$\boxed{\sum_{i=1}^{N} \text{ways}(i-1) \times \text{ways}(N-i)}$$

No. of BSTs : Bottom-Up

```
dp[N+1] = {0};
dp[0] = dp[1] = 1;

for(int i = 2; i <= N; i++){
      for(int j = 1; j <= i; j++){
            dp[i] += dp[j - 1] * dp[i - j];
      }
}
return dp[N];
```

No. of BSTs : Top-Down

```
int num_bsts(int N){
      if(N == 0 || N == 1) return 1;
      if(dp[N] != -1) return dp[N];
      dp[N] = 0;
      for(int i = 1; i <= N; i++){
            dp[N] += num_bsts(i - 1) * num_bsts(N-i);
      }
      return dp[N];
}
```

TopDown : Matrix Chain Multiplication

```
int minCost(int i, int j){
      if (i == j)
         return 0;
      if(dp[i][j] != -1) return dp[i][j];
      int ans = INT_MAX;
      for (int k = i; k < j; k++){
         temp = minCost(i, k) + minCost(k + 1, j)
                  + d[i - 1] * d[k] * d[j];

        ans = min(ans, temp);
     }
      dp[i][j] = ans;
      return ans;
}
ANS = minCost(1, d.size() - 1);
```