Will start at 7:35 am

Q ⇒ Given N arr[], find length of Longest incr subseq (LIS)

Note: $a_1 < a_2 < a_3$

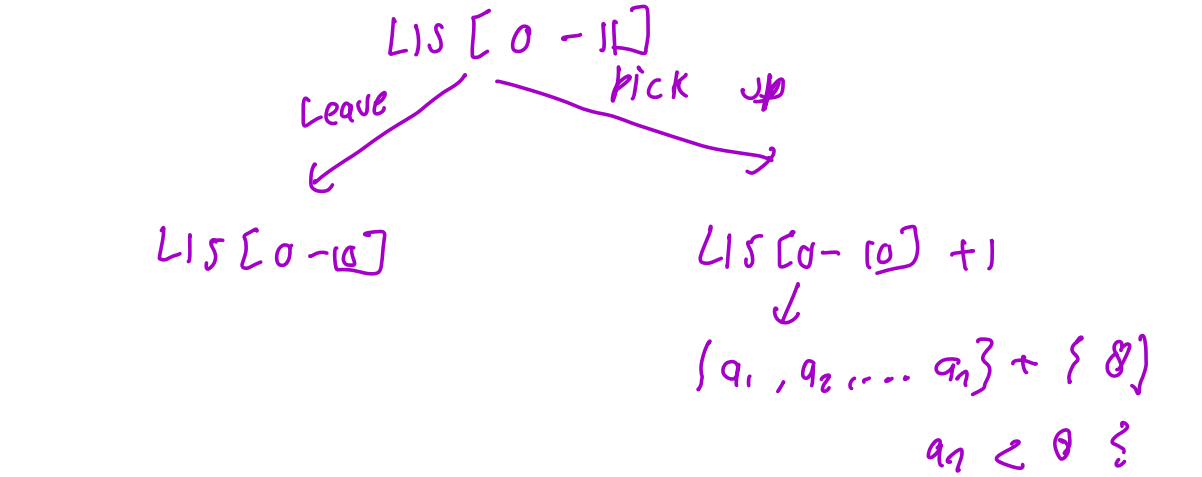
Ex 1: arr[5] = {9, 2, 4, 3, 10}

ans 1: {2, 4, 10} ⇒ length ⇒ 3
{2, 3, 10}

Ex 2: arr[6] = {2, -1, 6, 3, 7, 9}

len 4 ⇒ 2, 6, 7, 9

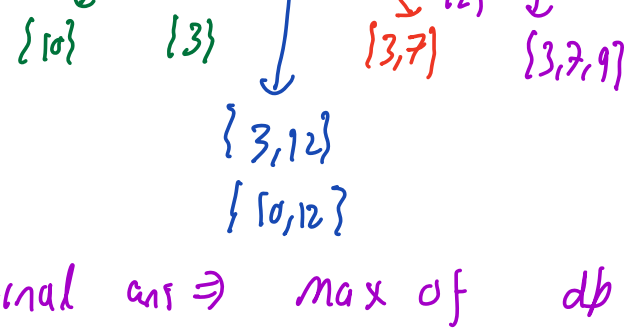
	0	1	2	3	4	5	6	7	8	9	10	11
arr[12] =	10	3	12	7	2	9	11	20	11	13	6	0



dp[10] ⇒

dp[i] = length of LIS from [0-i] ending at ith index → [This should definetly contain arr[i]]

	0	1	2	3	4	5	6	7	8	9	10	11
arr[12] =	10	3	12	7	2	9	11	20	11	13	6	0
dp[12] =	1	1	2	2	1	3	4	5	4	5	2	3



final ans ⇒ max of dp

$$dp[i] = \max_{j=0}^{i-1} \{ dp[j] \text{ if } arr[j] < arr[i] \} + 1$$

```
int LIS ( int arr [])
  {
     int n = arr.length;
     int dp[n];
     dp[0] = 1;
     for( i=1 ; i<n ; i++)
       {
          v=0;
         for( j= 0 ; j <i ; j++)
           {
              if (arr[j] < arr [i]
                   v = max ( v, dp[j]),
           }
          dp[i] = v+1 ;
       }
     return  maxElem ( dp);
  }
```

state * TC
every state

N * N

TC: $O(N^2)$

SC: $O(N)$

$O(n \log n)$

Q2 ⇒ N Houses

Given N houses & cost associated to color each house in R / G / B. Find min cost to paint all houses.

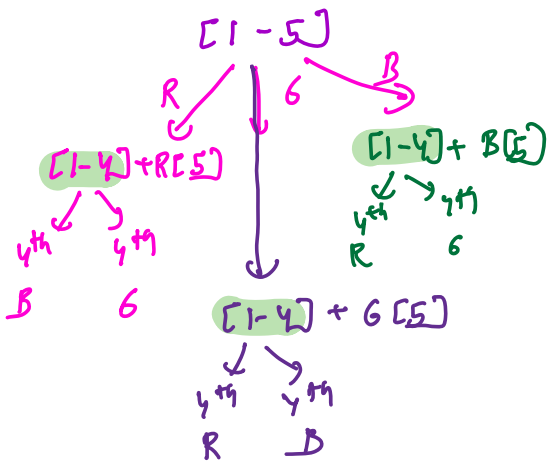
Note: No 2 adjescent house should have same color

N = 3 :

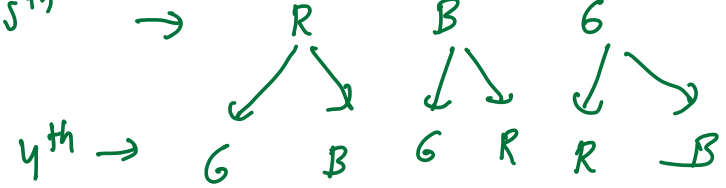
	1	2	3
R	5	8	4
G	2	1	5
B	6	9	7

B R G - 19
G R G - 15
R G R - 10
G G R - Invalid

	1	2	3	4	5
R	5	8	4	2	1
G	2	1	5	6	7
B	6	9	7	1	5



[1-5] :



Note: We should also keep track color we are ending with

R → 0 , B → 1 , G → 2

dp[i][0] = { min cost to paint [i-1]; s.t. ith house ends with Red }

dp[i][1] = { min cost to paint [i-1]; s.t. ith house ends with Blue }

dp[i][2] = { min cost to paint [i-1]; s.t. ith house ends with green }

DP expression

$$dp[i][0] = R[i] + \min(dp[i-1][1], dp[i-1][2])$$

$$dp[i][1] = B[i] + \min(dp[i-1][0], dp[i-1][2])$$

$$dp[i][2] = G[i] + \min(dp[i-1][0], dp[i-1][1]);$$

Base conditions

$dp[0][0] = dp[0][1] = dp[0][2] = 0$

DP Table : $dp[N+1][3]$

Pseudo Code

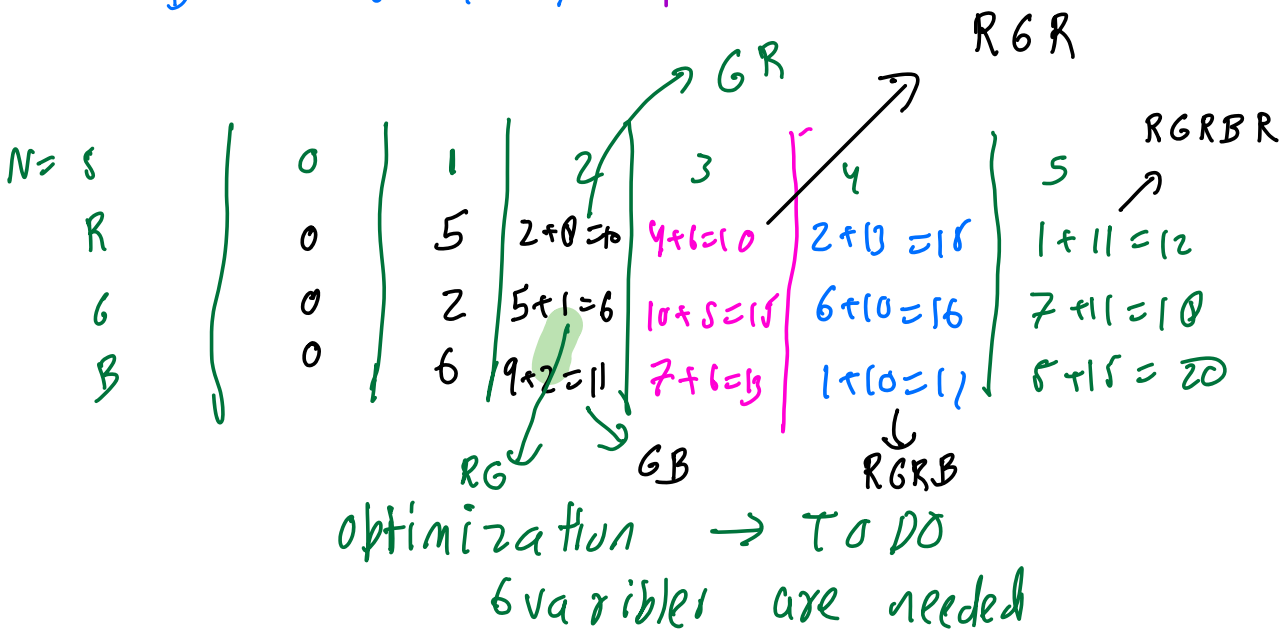
R[i] = cost to paint ith house in Red color
G[i] = ——— green ———
B[i] = ——— Blue ———

```
int dp[N+1][3]
dp[0][0] = dp[0][1] = dp[0][2] = 0

                                TC: O(N)
i=1 ; i<=n ; i++                SC: O(N)
{
  dp[i][0] = R[i] + min(dp[i-1][1],
                        dp[i-1][2])
  dp[i][1] = B[i] + min(dp[i-1][0],
                        dp[i-1][2])
  dp[i][2] = G[i] + min(dp[i-1][0],
                        dp[i-1][1]);
}
return min(dp[N][2], dp[N][0], dp[N][1]);
}
```

)

	1	2	3	4	5
R	5	8	4	2	1
G	2	1	5	6	7
B	6	9	7	1	5



optimization → TO DO

6 variables are needed