Number Of Longest Increasing subsequance.

arr: { 1, 5, 4, 3, 2, 6, 7, 10, 8, 9 }

dp[ ]	1	2	2	2	2	3	4	5	5	5
count[ ]	1	1	1	1	1	4	4	4	4	4

1, 6

1, 5, 6 → 3

1, 4, 6 → 4 +1

1, 3, 6 → 5

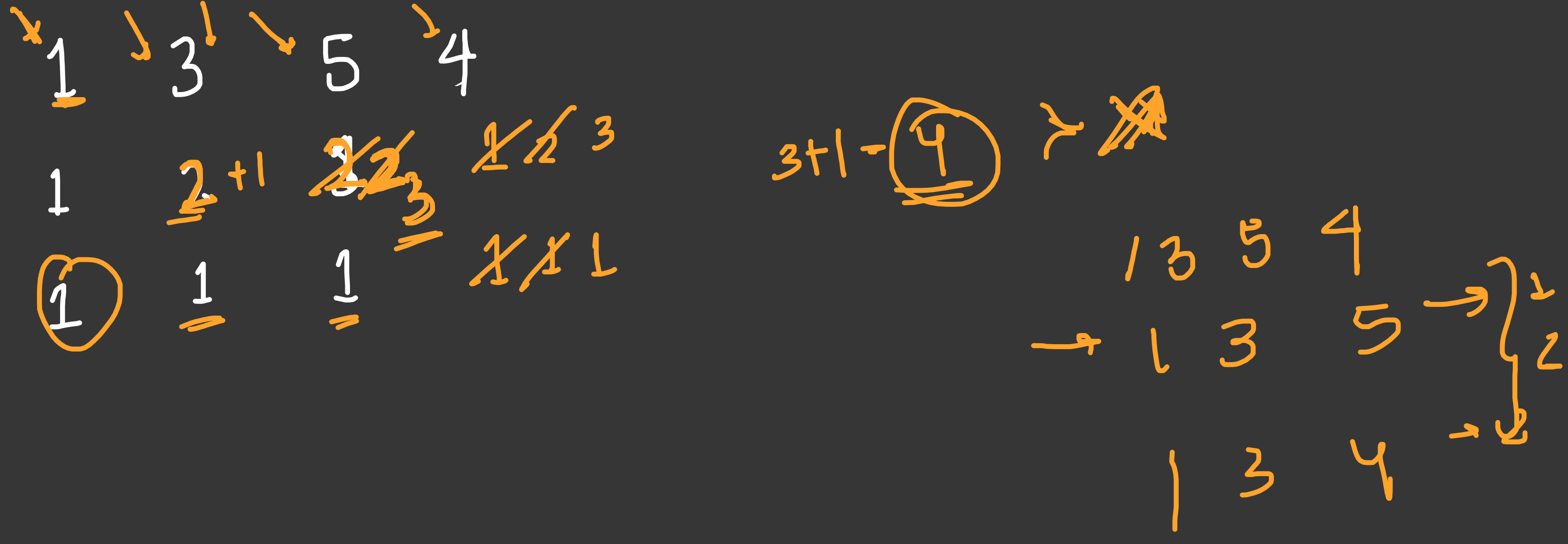
1, 2, 6

1 5 6 7 10

1 4 6 7 10

1 3 6 7 10

1 2 6 7 10



1 3 5 4
1 2 +1
3+1 = 4
1 3 5 4
1 3 5
1 3 4

Pseudocode

```
for (i=1; i < n; i++)
    for (j=0, j < i; j++)
        if( arr[i] > arr[j] && 1+dp[j] >= dp[i])
            dp[i] = 1+dp[j],
            ct[i] = ct[j],
        else if( arr[i] > arr[j] && 1+dp[j] == dp[i])

            ct[i] += ct[j]
```

Dry run;

1 3 5 4

dp: | 1 | 2 | 3 | 3 |

ct: | 1 | 1 | 1 | 1 |

i = 1

j = 0 → 1

3 > 1 1 + dp[1+1 > 1

i = 2

j = 0 → 2

5 > 1 1 + dp[1 > 1

j = 1

5 > 3 1+2 3 > 2

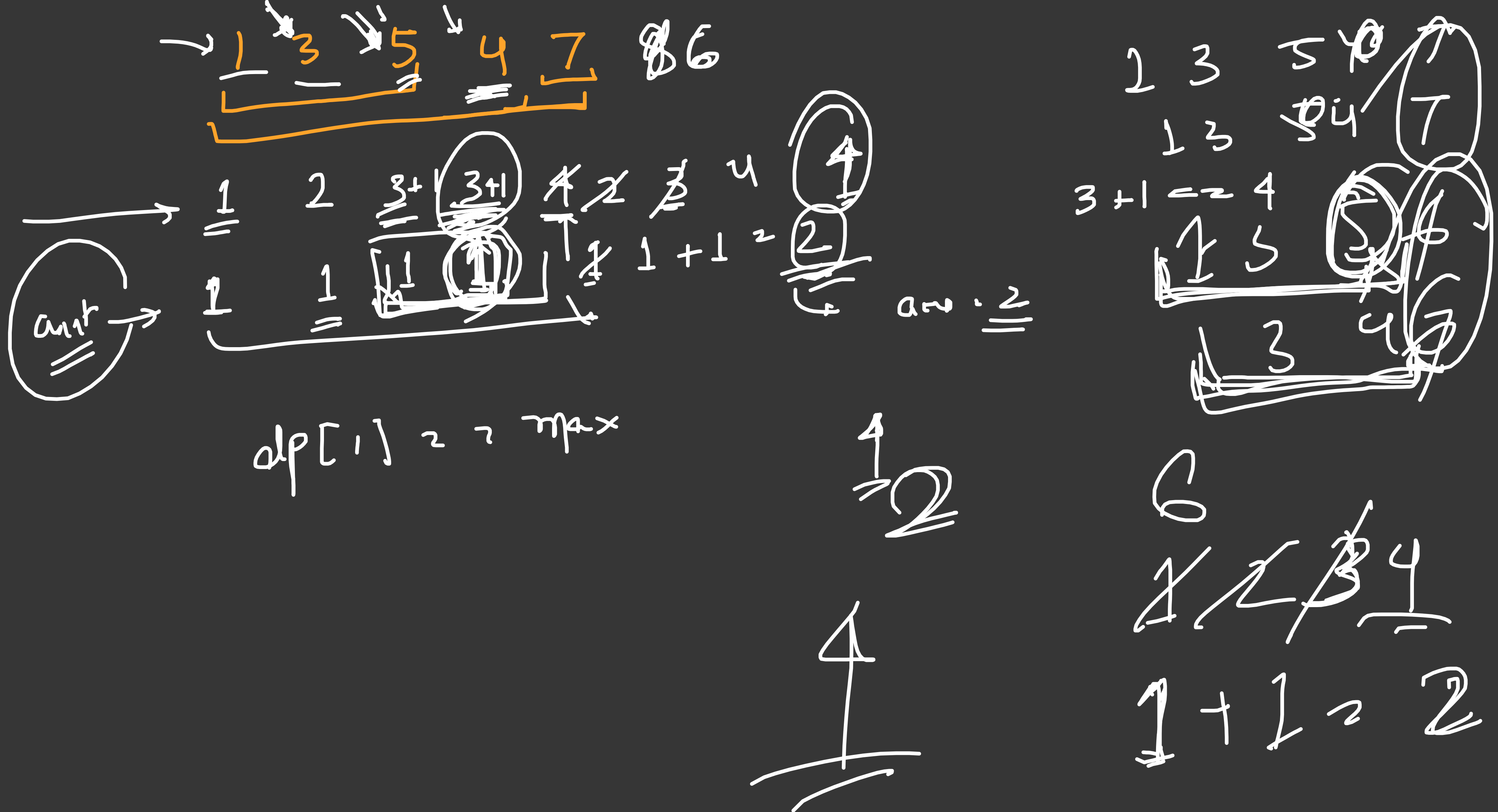
i = 3

j = 0 → 3

j = 0 4 > 1

j = 1 3 4 > 3 1+2

j = 2 5 < 4


1 3 5 4 7
ans = 2
dp[1] = max
3+1 == 4
1+1 = 2