3-4 hr

30 mins

Recordy

Notes

Code

Explain

15 min

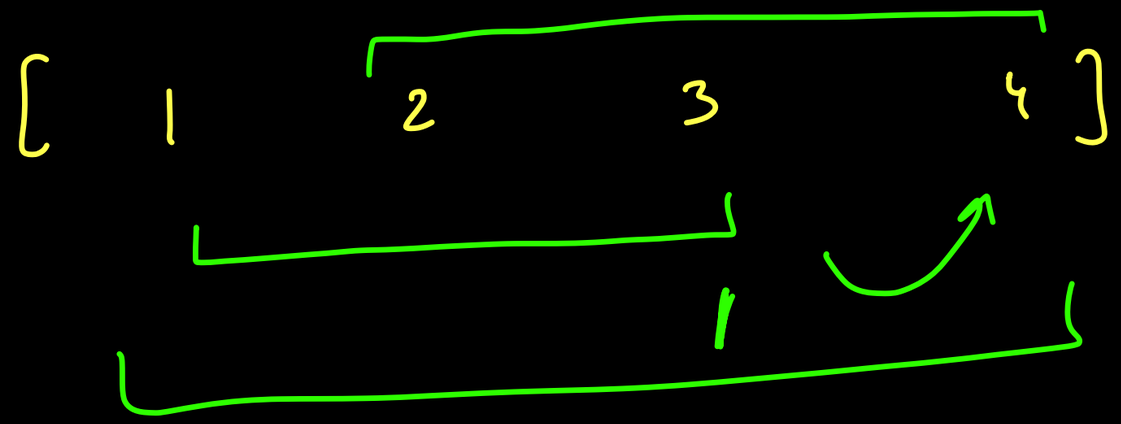
35 min → 100 min

10 min

Team

Thr

K.T → Knowledge Transfer



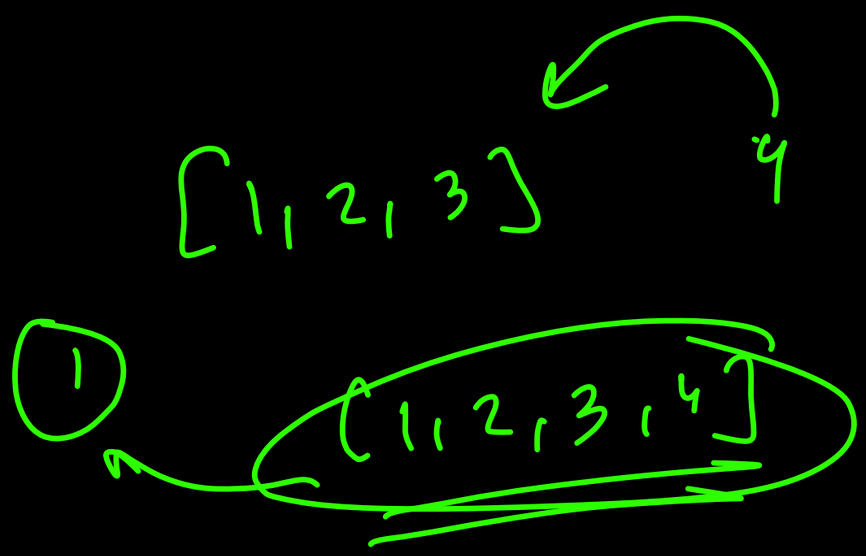
→ LIS

↓

longest Inc Subsq

(2-1) == (3-2)

(4-3) == (3-2)



[1, 2, 3, 4] ⟶ (1, 2, 3) (2, 3, 4) (1, 2, 3, 4)

atleast 3 length

(4, 5, 6)

i

0 1 2 3 4 5

[1, 2, 3, 4, 5, 6]

1 1+1 1+2 1+3

2, 3, 4, 5
(1, 2, 3, 4, 5

3, 4, 5
2, 3, 4, 5
(1, 2, 3, 4, 5

6
5
4

arr[i] - arr[i-1] == arr(i-1) - arr(i-2)

(i, i-1, i-2)

1, 2, 3, 4, 7, 8, 9

→ 0 0, 1, 2, 0, 0, 1

(1, 2, 3)

(2, 3, 4)

(1, 2, 3, 4)

(7, 8, 9)

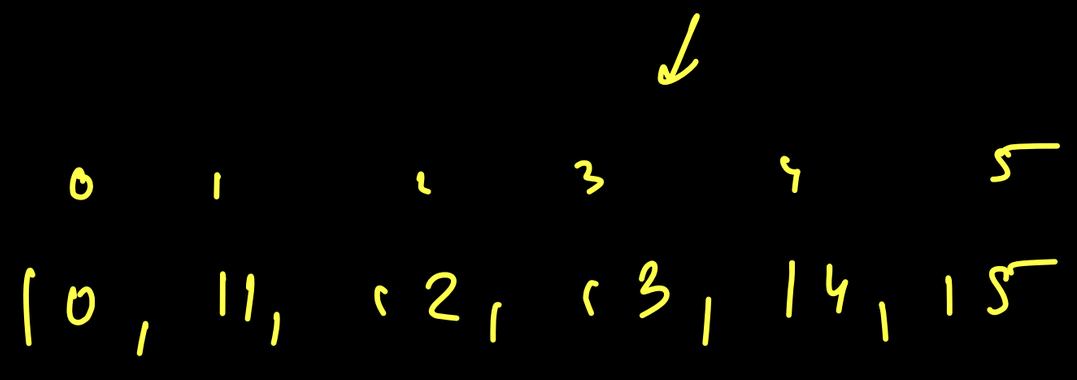
$f(i)$ = no. of arithmetic Subarrays ending at i.

$$f(i) = \begin{cases} 0 & \text{if } a[i]-a[i-1] \neq a[i-1]-a[i-2] \\ 1 + f(i-1) & \text{else} \end{cases}$$

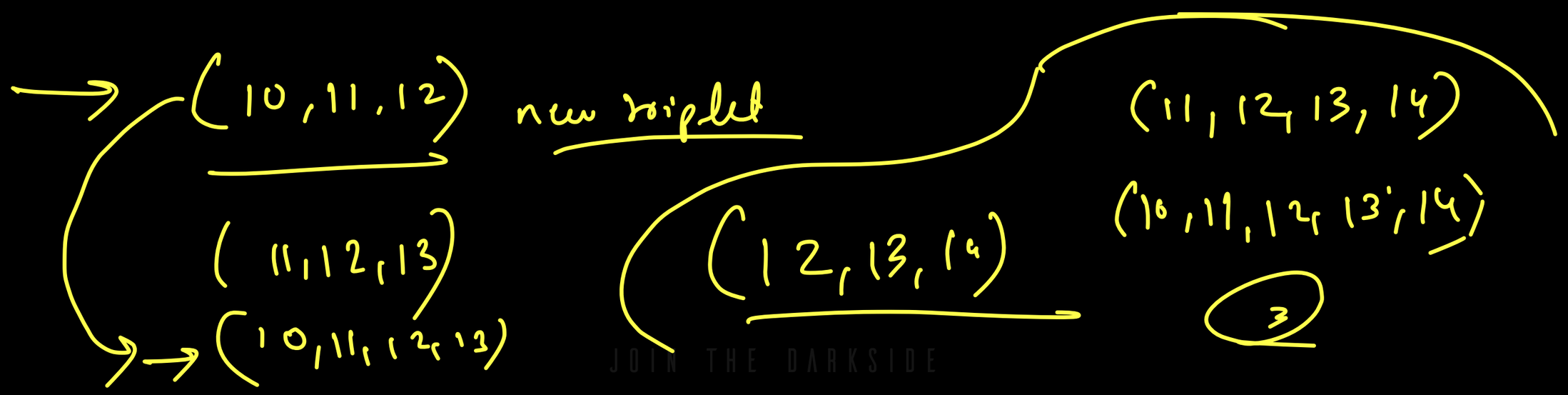
Brand new triplet $(i, i-1, i-1)$

because we can attach i^{th} element to all subarrays ending at $(i-1)$

$$ans \rightarrow \sum_{i=0}^{n-1} f(i)$$



$f(i) \rightarrow 0 \quad 0 \quad 1 \quad 2 \quad 3$



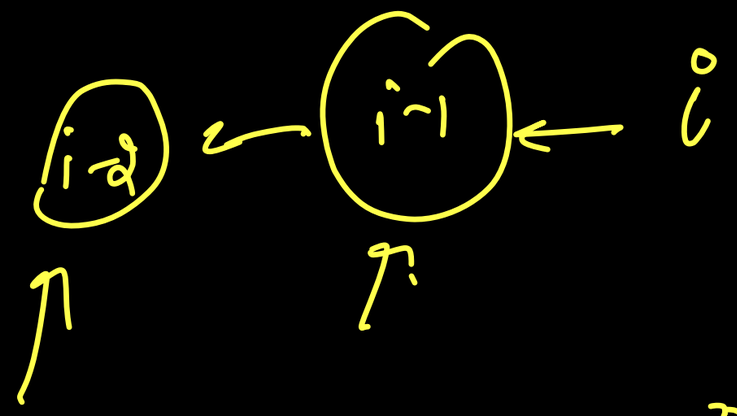
$i=0 \rightarrow f_0$

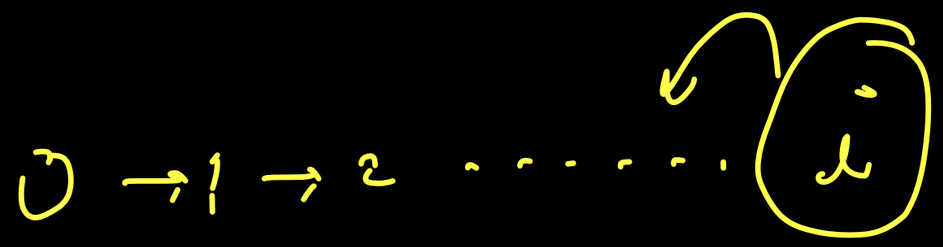
$i=1 \quad f_1$

$i=2 \rightarrow f_2 \rightarrow f_1$

$i=3 \quad f_3 \rightarrow f_2 \rightarrow f_1$

$i=4 \quad f_4 \rightarrow f_3 \rightarrow f_2 \rightarrow f_1$





atleast 3 elements

ending at i

[2, 4, 6, 8, 10] → lis

[3, 6, 7, 9]

3, 6, 9

[2, 4, 6]

(4, 6, 8) (2, 4, 6, 8)

(10, 8, 6) (2, 4, 6, 8, 10) (4, 6, 8, 10)

(10, 6, 2)

current subseq & feature optn

(4,5,6) — (2,4,6)

(4,5,6,7)

(5,6,7)

(5,6,7,8)

(6,7,8)

(4,5,6,7,8)

(2, 4, 5, 6, 7, 8, 10)

(2-1) (1-1) (1-2) (1-3) (1-4) (2-4)

(3-1) (2-2) (2-1) (2-3) (4-2)

(4-1) (3-1) (3-2) (5-1)

(5-1) (4-1) (6-1)

(6-1) (8-1)

(3-2)

am20 +1 +1

+2 +3

+2 +1

+3 +1

+1 +1

1 potential subseq

well (mon diff)

(4,6,8)

(2,4,6,8)

(6,8)

(5,8)

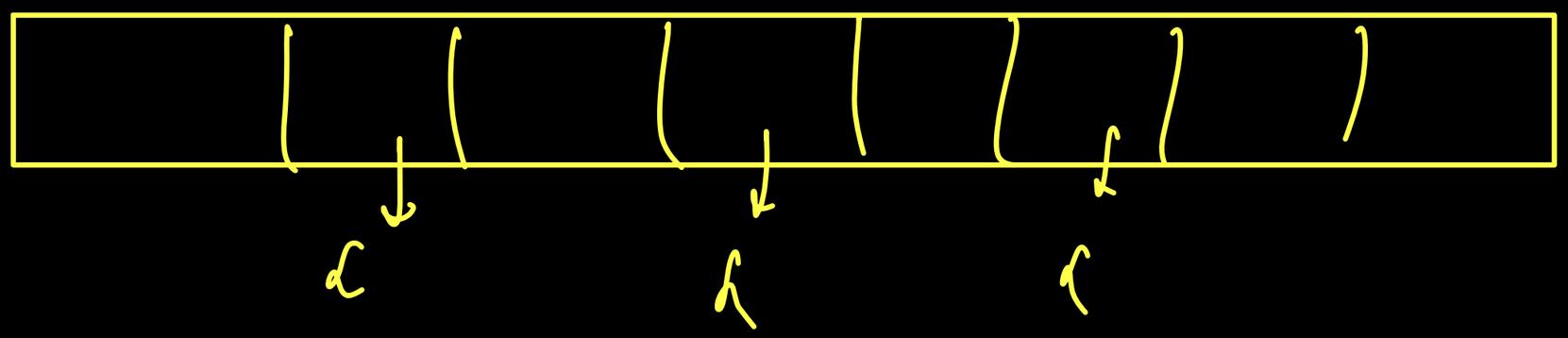
(4,5,8)

dp - memo

ending at every index we store all the subsequences (arithmetic & non-arithmetic) with a given common diff

For every value of i → <key, value>

diff → all possible subseq



Python → array of dict

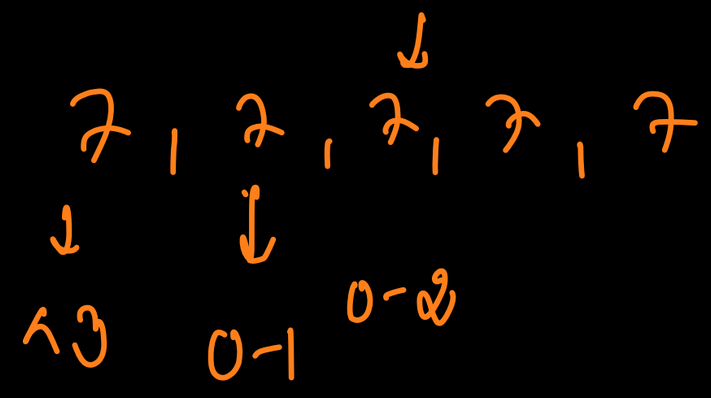
JS → array of object

Java → array of hashmap

C++ → array of unordered_map

```
ans = 0
for (i = 1;  i < n ;  i++) {
    for (j = 0 ; j < i ; j++) {
        diff = arr[i] - arr[j]
        if (dp[j].get(diff)) {
            ans += dp[j].get(diff).
            dp[i].put(diff, dp[j].get(diff)
                                  +1)
        }
    }
}
```

```
for ( i=1 ; i<n ; i++)
    for (j=0 ; j<i ; j++)
        diff = a[i] - a[j]
        if (dp[j][diff])
        ans += dp[j][diff]
        dp[i][diff] = 1+dp[j][diff]
    }
}
```



$a_m = 0+1$