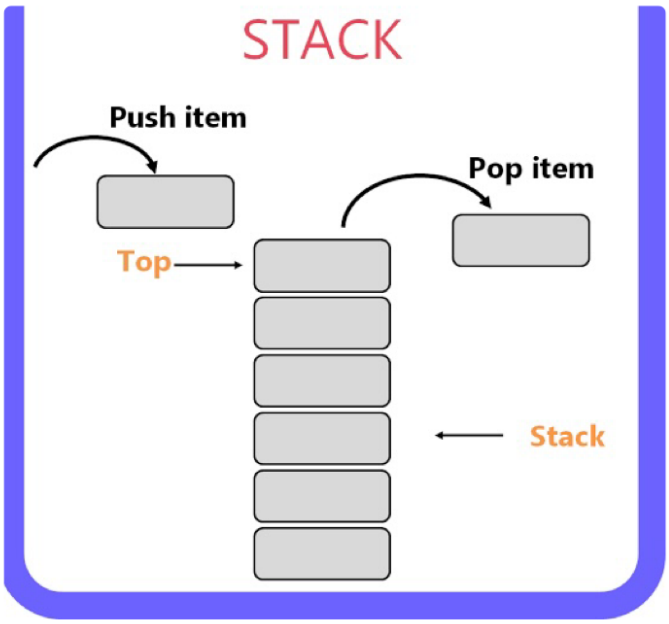

STACK
video - 13



- Push
- Pop
- Top

Leetcode 456 Medium

Easy (3 Approach)

B → Better approach

132 Pattern

Company :- amazon Meta

456. 132 Pattern

Medium 6151 341 Add to List Share

Given an array of `n` integers `nums`, a **132 pattern** is a subsequence of three integers `nums[i]`, `nums[j]` and `nums[k]` such that `i < j < k` and `nums[i] < nums[k] < nums[j]`.

Return `true` *if there is a* ***132 pattern*** *in* `nums`, *otherwise, return* `false`.

Example :- nums = { 3, 1, 4, 2 }

Output = True.

nums = {-1, 3, 2, 0}

Output = True. -1 < 2 < 3

i, j, K

Brute Force :-

nums[i] < nums[k] < nums[j]

→ for (int i=0 ; i<n-2; i++)

for (int j= i+1 ; j<n-1 ; j++) {

$O(n^3)$

TLE.

```
if(nums[i] < nums[j]) {
    for(int K = j+1; j<i ) {
        if(nums[i] < num[K]
            && num[K] < nums[j])
            ref True.
    }
}
```

Better Approach

(i < j < K)

nums [i] < nums[K] < nums[j]

smallest

```
                    j   K
nums = { 3,  1,   4   2 }
         0   1    2   3
                  ↑
```

$O(n^2)$.

```
num-i = nums[0]; // ∄ 1

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

    num-i = min(num-i, nums[j]);
```

```
for( k = j+1; k<n; k++) {
    if( num_i < num[k] && nums[k] < nums[j])
        return True;
}
```

~~n log(n) → sorting~~

O(n)

Best Approach

- ⇒ why (n-1) to 0 traversal
- ⇒ Why Stack ???

↳ Dependent for loops

↳ Monotonic Stack.

i < j < k

num1 num2 num3

largest element

-1 ∅ 2 ∅ 2 3

largest ∅ 2 3

{ -1 , 3 , 2 , 0 }

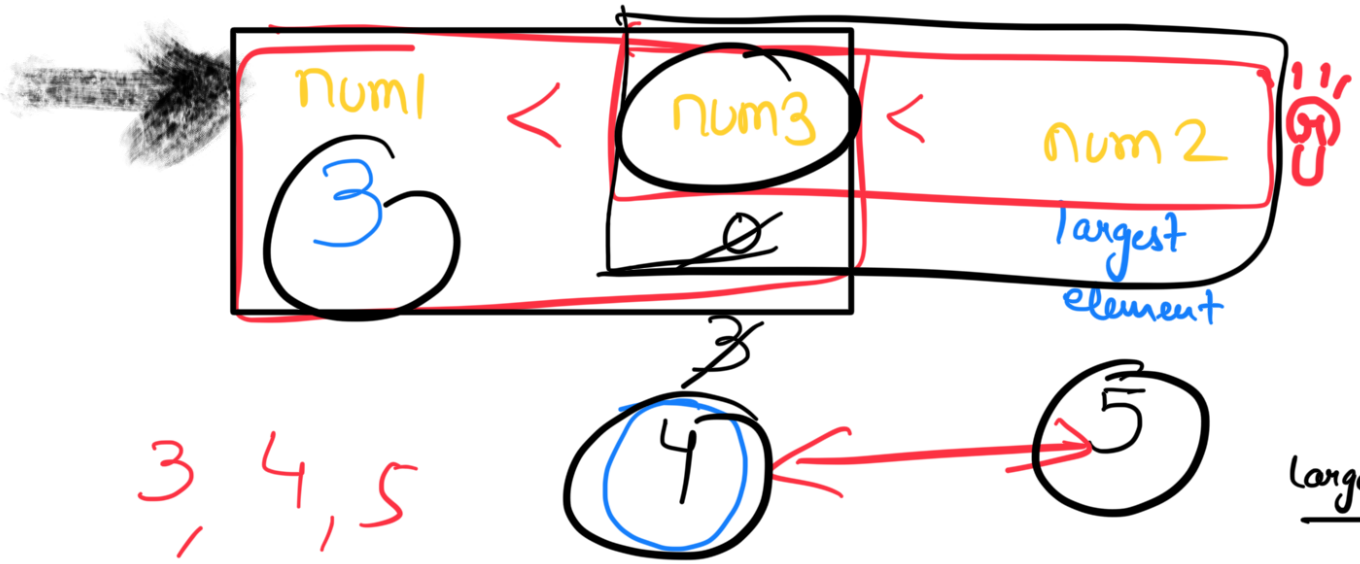
i i i i

if (nums[i] < num3)
return True;

Another Good Example Dry Run :-

i < j < k

num1 num2 num3



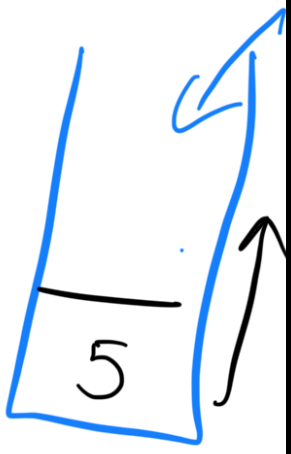
3, 4, 5

Largest

{ 3, 5, 0, 3, 4 }

i i i i i

0 1 2 3 4



i < j < k

3 < 4 < 5

num3 = −∞

if (nums[i] < num3)

Yeh True;