lambda function

{1, 2, 10, -12, 3, 9}

Integer Integer

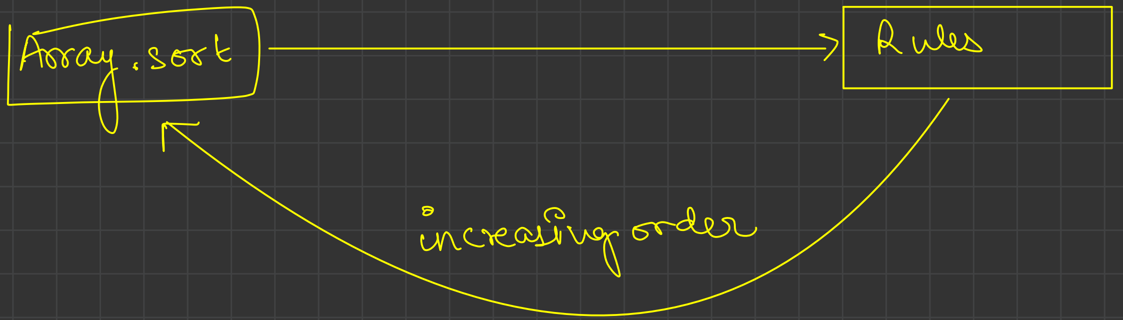
```
Arrays.sort(arr, (a, b) -> {

});
```

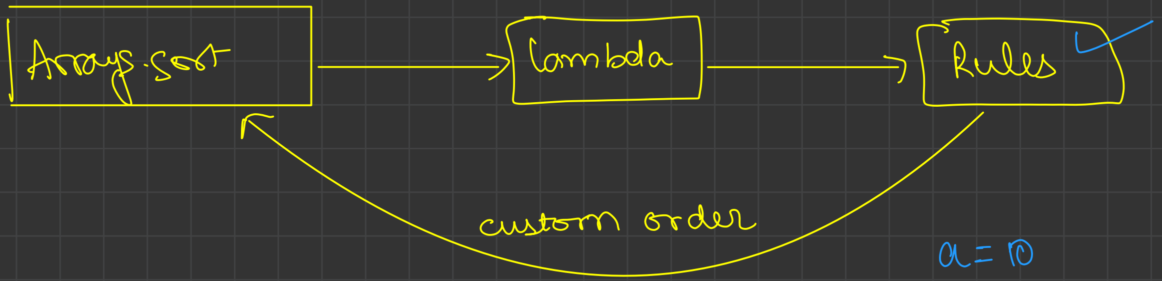
return

-ve No. → smaller

+ve No. → larger

0 → equal





a = 10
b = 20

a < b → (a − b) → (−)ve → move left

a > b → (a − b) → (+)ve → move right

(10, 20)
a b

b, a
(10, 20)

a = 20
b = 10

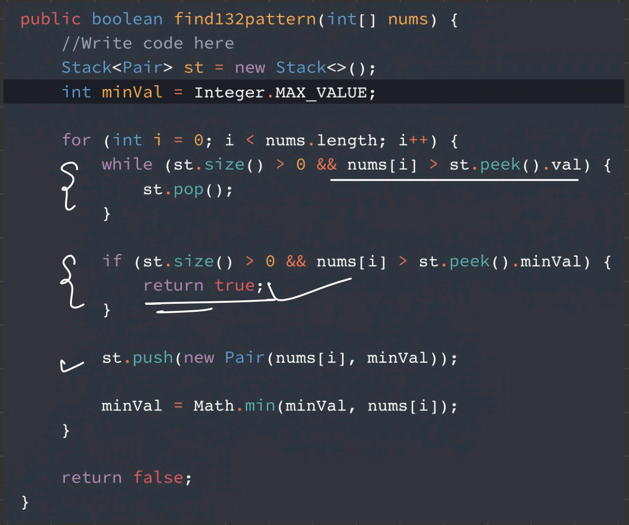
a = 10, b = 20 → dec

a < b → (b − a) → (+)ve → move right → (b, a) (20, 10)

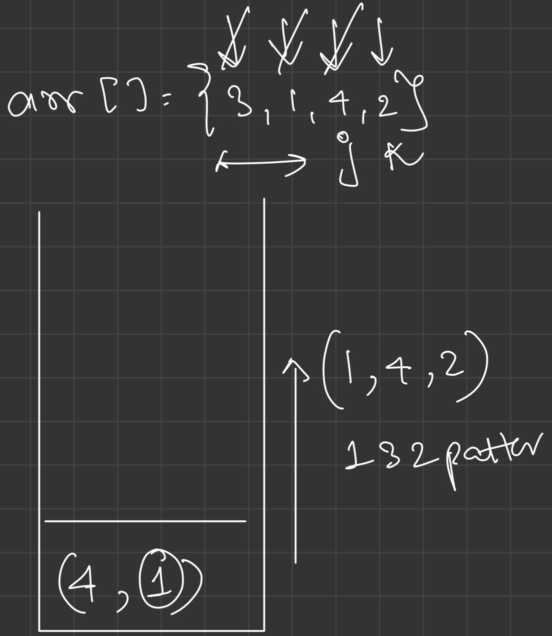
a > b → (b − a) → (−)ve → move left → (a, b) (20, 10) → decreasing

a = 20, b = 10

132 pattern.



```java
public boolean find132pattern(int[] nums) {
    //Write code here
    Stack<Pair> st = new Stack<>();
    int minVal = Integer.MAX_VALUE;

    for (int i = 0; i < nums.length; i++) {
        while (st.size() > 0 && nums[i] > st.peek().val) {
            st.pop();
        }

        if (st.size() > 0 && nums[i] > st.peek().minVal) {
            return true;
        }

        st.push(new Pair(nums[i], minVal));

        minVal = Math.min(minVal, nums[i]);
    }

    return false;
}
```



TC: O(N)
SC: O(N)

minVal = ~~∞~~ ~~3~~ 1

design HashSet.