<https://leetcode.com/problems/removing-minimum-and-maximum-from-array/description/>

You are given a 0-indexed array of distinct integers nums.

There is an element in nums that has the lowest value and an element that has the highest value. We call them the minimum and maximum respectively. Your goal is to remove both these elements from the array.

A deletion is defined as either removing an element from the front of the array or removing an element from the back of the array.

Return the minimum number of deletions it would take to remove both the minimum and maximum element from the array.

**Example 1:**

**Input:** nums = [2,**10**,7,5,4,**1**,8,6]

**Output:** 5

**Explanation:**

The minimum element in the array is nums[5], which is 1.

The maximum element in the array is nums[1], which is 10.

We can remove both the minimum and maximum by removing 2 elements from the front and 3 elements from the back.

This results in 2 + 3 = 5 deletions, which is the minimum number possible.

**Example 2:**

**Input:** nums = [0,**-4**,**19**,1,8,-2,-3,5]

**Output:** 3

**Explanation:**

The minimum element in the array is nums[1], which is -4.

The maximum element in the array is nums[2], which is 19.

We can remove both the minimum and maximum by removing 3 elements from the front.

This results in only 3 deletions, which is the minimum number possible.

**Example 3:**

**Input:** nums = [**101**]

**Output:** 1

**Explanation:**

There is only one element in the array, which makes it both the minimum and maximum element.We can remove it with 1 deletion.

**Constraints:**

1 <= nums.length <= 10^5

-10^5 <= nums[i] <= 10^5

The integers in nums are **distinct**.

**Attempt 1: 2024-12-26**

**Solution 1: Math (10 min)**

class Solution {

    public int minimumDeletions(int[] nums) {

        int max = -100001;

        int min = 100001;

        int max\_index = -1;

        int min\_index = -1;

        for(int i = 0; i < nums.length; i++) {

            if(max < nums[i]) {

                max = nums[i];

                max\_index = i;

            }

            if(min > nums[i]) {

                min = nums[i];

                min\_index = i;

            }

        }

        int index\_1 = -1;

        int index\_2 = -1;

        if(max\_index > min\_index) {

            index\_1 = min\_index;

            index\_2 = max\_index;

        } else {

            index\_1 = max\_index;

            index\_2 = min\_index;

        }

        // Take nums = {2,10,7,5,4,1,8,6} as example

        // index\_1 is for max\_index = 1

        // index\_2 is for min\_index = 5

        // 'op\_count\_1' is operation count before reach index\_1 as 1 (remove 2)

        // 'op\_count\_2' is operation count after index\_2 and before reach index\_2 as 3 (remove 7,5,4)

        // 'op\_count\_3' is operation count after index\_2 as 2 (remove 8,6)

        // The concept is we only have three choices (op\_count\_1 + op\_count\_2,

        // op\_count\_1 + op\_count\_3, op\_count\_2 + op\_count\_3), which means

        // remove the smallest sum up two out of 'op\_count\_1', 'op\_count\_2'

        // and 'op\_count\_3', then remain part is only need to take out target

        // two elements (minimum  + maximum), so when we find the smallest

        // combinations out of three choices, plus 2 is the answer

        int op\_count\_1 = index\_1;

        int op\_count\_2 = index\_2 - index\_1 - 1;

        int op\_count\_3 = nums.length - index\_2 - 1;

        return 2 + Math.min(op\_count\_1 + op\_count\_2, Math.min(op\_count\_1 + op\_count\_3, op\_count\_2 + op\_count\_3));

    }

}

Time Complexity: O(n)

Space Complexity: O(1)

**Refer to**

<https://leetcode.com/problems/removing-minimum-and-maximum-from-array/solutions/1599809/c-only-3-cases/>

**Intuition**: Find the two indices of the maximum and minimum elements, say a and b. The order doesn't matter. We only need to consider 3 cases:

Delete both from left and right.

Delete only from left

Delete only from right

**Algorithm**:

Make sure a <= b, and return the minimum of the following:

(a + 1) + (N - b) -> delete both from left and right

b + 1 -> delete from left

N - a -> delete from right

// OJ: https://leetcode.com/problems/removing-minimum-and-maximum-from-array/

// Author: github.com/lzl124631x

// Time: O(N)

// Space: O(1)

class Solution {

public:

int minimumDeletions(vector<int>& A) {

int a = max\_element(begin(A), end(A)) - begin(A), b = min\_element(begin(A), end(A)) - begin(A), N = A.size();

if (a > b) swap(a, b);

return min({ a + 1 + N - b, b + 1, N - a });

}

};

**Refer to**

[L1423.Maximum Points You Can Obtain from Cards (Ref.L1658)](note://WEBb70d72cb469c39161ffaeb2d08c64a44)

[L1647.Minimum Deletions to Make Character Frequencies Unique](note://WEB8a683715985e8e8ffa6c1edb2a9f6273)