

Finding Minimum Maximum in Array

Course on C-Programming & Data Structures: GATE - 2024 & 2025

Data Structure: Array: Min Max

By: Vishvadeep Gothi



Hello!

I am Vishvadeep Gothi

I am here because I love to teach

Invertⁿ

always
at end



$\Theta(1)$

R.T.
complexity

based on
index



$O(n)$

run-time
complexity

Deletⁿ in array

always
at end



$\Theta(1)$

based on
index



$O(n)$

Finding Minimum Element

9, 15, 12, 6, 10, 5, 4, 8, 2, 3

Min = ~~9 15 12 6 10 5 4 8 2~~

for n elements in array ↓

n-1 Comparisons needed

alg o

min = A[LB]

for k = LB + 1 to UB, k++

{ if (A[k] < min)

{ min = A[k];

} } return min;

R.T. complexity

$\Rightarrow \Theta(n)$

Space complexity $\Rightarrow \Theta(1)$

Finding Maximum Element

```
max = A[0]
for k = 1 to n, k++
{
    if (A[k] > max)
        max = A[k]
}
return max
```

```
max = A[0];
for (k = 1; k < n ; k++)
{
    if (A[k] > max)
        max = A[k];
}
return max;
```

No. of comparisons needed
to find max of n elements = $n - 1$

R.T. Complexity = $\Theta(n)$

Space = 1 = $\Theta(1)$

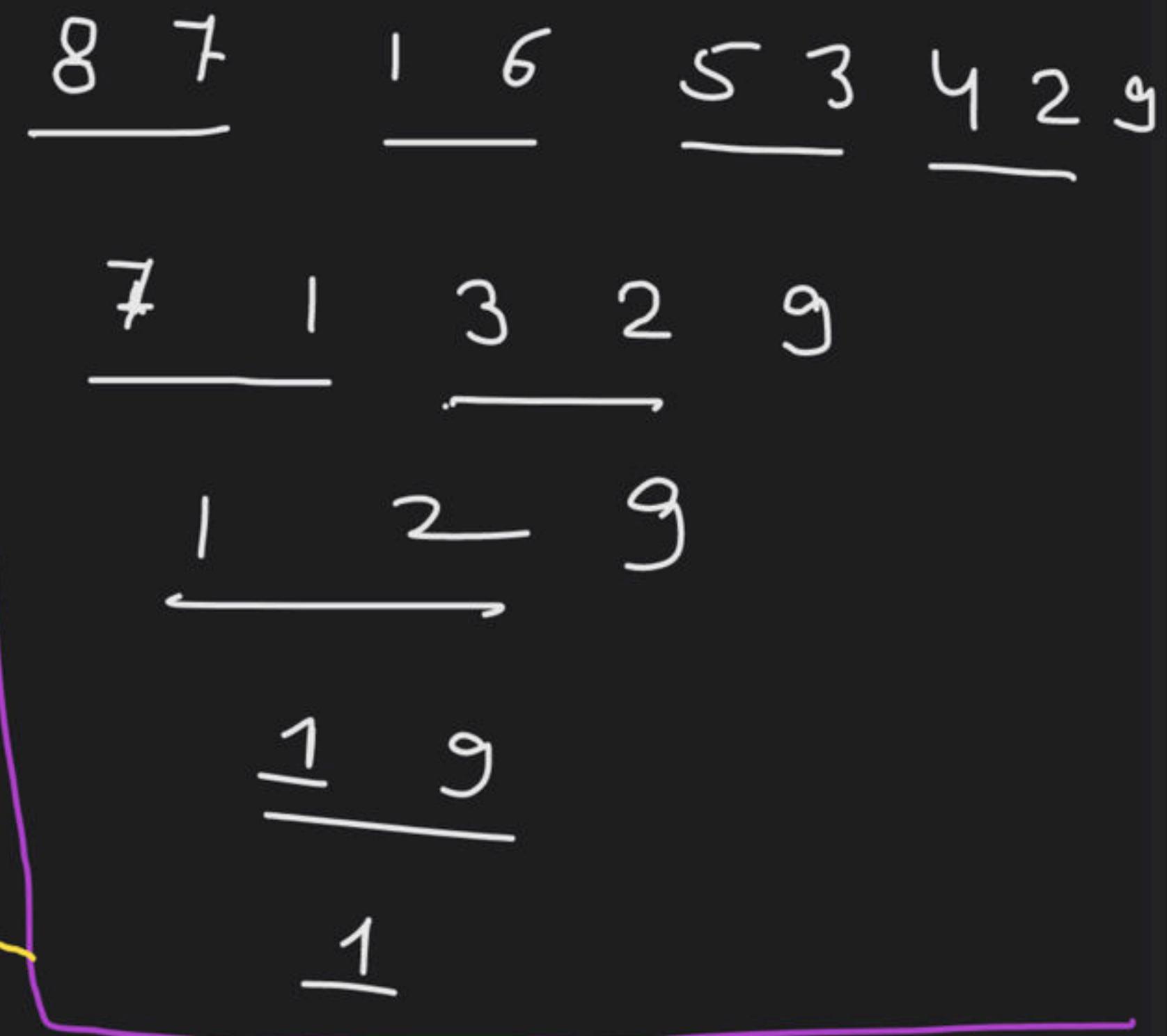
Tournament Method:-



no. of comparisons = $n - 1$

Space complexity = $\Theta(n)$

Run time complexity = $\Theta(n)$



Finding Minimum & Maximum Both

Method 1 :-

- ① Find min
- ② Find max

	no. of comparisons
① Find min	$n-1$
② Find max	$n-1$
Total	$2n - 2$ comparisons

Method 2:-

1 2 3 4 5 6 7 8

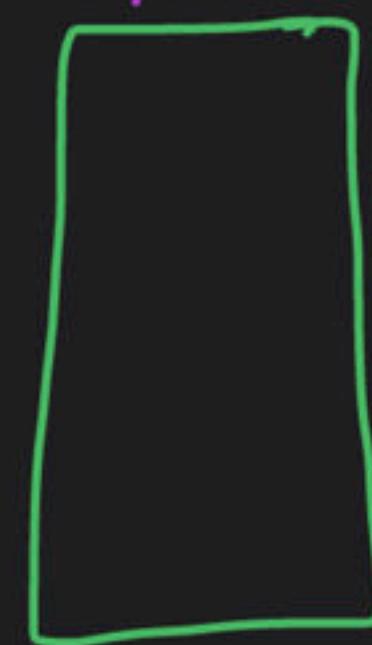
a_1, a_2 , a_3, a_4 , a_5, a_6 , ..., a_{n-1}, a_n

$n/2$ element



lesser
element
array

$n/2$ elements



greater
element
array

No. of Comparisons

To make 2 lists $\Rightarrow \frac{n}{2}$

To find min $\Rightarrow \frac{n}{2} - 1$

To find max $\Rightarrow \frac{n}{2} - 1$

$$\text{Total} = \left\lceil \frac{3n}{2} - 2 \right\rceil$$

$$= 1.5n - 2$$

if n is odd =>

$q_1 \ q_L$ $q_3 \ q_4$ $q_{n-2} \ q_{n-1}$ q_n

then last element is kept into both list.

$$\text{No. of Comparisons} = \left\lceil \frac{3n}{2} \right\rceil - 2$$

Ans = 148

Question GATE-2014

The minimum number of comparisons required to find the minimum and the maximum of 100 numbers is _____?

$$\frac{3 * 100}{2} - 2 = 148$$

Finding Second Minimum

Method 1:-

1. Find min of n elements

$\Rightarrow n-1$

2. ————— ||— remaining $n-1$ elements

$n-2$

Total

$2n-3$ Comparisons

Comparisons

Method 2:-

- ① Find min with tournament method
- ② Make a list of all elements which are compared with min element, find min. of this list

Comparison

$n - 1$

$\log_2 n - 1$

$$\text{Total Comparisons} = n + \lceil \log_2 n \rceil - 2$$

Searching in Array

1. Linear Search
2. Binary Search

Linear Search

Search for a target value in an array.

Time complexity: $O(n)$.

Space complexity: $O(1)$.

Implementation:

```
function linearSearch(arr, target) {
```

```
    for (let i = 0; i < arr.length; i++) {
```

```
        if (arr[i] === target) {
```

```
            return i;
```

```
    }
```

```
    return -1;
```

```
}
```

Example:

```
const arr = [1, 2, 3, 4, 5];
```

```
const target = 3;
```

```
const index = linearSearch(arr, target);
```

```
console.log(index); // Output: 2
```

Note: This search algorithm is not efficient for large arrays.

Efficient search algorithms like binary search are available for sorted arrays.

Implementation:

```
function binarySearch(arr, target) {
```

```
    let left = 0;
```

```
    let right = arr.length - 1;
```

```
    while (left <= right) {
```

```
        const mid = Math.floor((left + right) / 2);
```

```
        if (arr[mid] === target) {
```

```
            return mid;
```

```
        } else if (arr[mid] < target) {
```

```
            left = mid + 1;
```

```
        } else {
```

```
            right = mid - 1;
```

```
        }
```

```
    }
```

```
    return -1;
```

```
}
```

Example:

```
const arr = [1, 2, 3, 4, 5];
```

```
const target = 3;
```

```
const index = binarySearch(arr, target);
```

```
console.log(index); // Output: 2
```

Note: This search algorithm is efficient for sorted arrays.

Implementation:

```
function search(arr, target) {
```

```
    const result = arr.findIndex(item => item === target);
```

```
    return result;
```

```
}
```

Example:

```
const arr = [1, 2, 3, 4, 5];
```

```
const target = 3;
```

```
const index = search(arr, target);
```

```
console.log(index); // Output: 2
```

Note: This search algorithm is efficient for arrays.

Implementation:

```
function search(arr, target) {
```

```
    const result = arr.indexOf(target);
```

```
    return result;
```

```
}
```

Binary Search

Question

Consider a sorted array of size n with duplicate elements. You have been given an element k , what is the time complexity to find the frequency of element k in the array?

- (A) $\theta(1)$
- (B) $\theta(\log n)$
- (C) $\theta(n)$
- (D) None

Question

Consider a sorted array of size n with duplicate elements. You have been given an element k , what is the time complexity to find that the element k is appeared atleast $n/3$ times or not?

- (A) $O(1)$
- (B) $O(\log n)$
- (C) $O(n)$
- (D) None

Question

Consider a sorted array of size n with duplicate elements. Find the time complexity to find whether an element is appeared more than $n/2$ times or not in the array?

- (A) $O(1)$
- (B) $O(\log n)$
- (C) $O(n)$
- (D) None



DPP

Lanes

Racing track with 5 tracks

25 horses

No watches or clocks

find 2 best horses by races

min. no. of races to be conducted.



Question 1

Consider you are the commander in chief for march past event and to have a good uniform arrangement in the march past, you need to arrange the students in a very specific manner. To identify this arrangement, you want to find the height difference of the students against each other. If there are n number of students and if it takes 40 seconds of time to compare heights of 2 students then, total time required to get the maximum of the differences between the heights of any 2 students for $n = 200$ is?

Question 2

Consider an array of n elements. Rotate the array in left k positions where k is a non-negative integer?

Question 3

Consider an array of n elements. Total comparisons required to calculate second minimum element of array if $n = 64$?

Question 4

Consider an array of size $n-1$ elements which contains elements between 1 to n (all unique) but 1 number is missing. Find the missing number?

Question 5

Consider an array which contains n elements. Arrange the array elements in such a way that all zeros in array are moved in the starting of the array, while maintaining the relative order of other elements.

Example:

Input: 1, 5, 7, 0, 2, 0, 4, 8, 0

Output: 0, 0, 0, 1, 5, 7, 2, 4, 8

Question 6

Consider an array A[-6 15], which is stored in memory starting from location 1000. Assume each element in memory is stored on 4 locations. Then the location of element A[2] is?

Question 7

There are n students in the class and the teacher is going to find the leader of the class. The teacher applies a simple strategy to find the leader as follows:

Teacher asked all students to stand in a single queue then given them numbering from 1 to n , starting from first student to last student. Then teacher has removed all students with odd numbering i.e. 1, 3, 5

For remaining students teacher given numbering once again from 1 without changing their relative order and then removed the odd number students again from the queue. Teacher repeated the same till there is only 1 student remaining in the queue. That one student becomes leader of the class.

Assume you want to be the leader then in the initial queue of n students on which position you will stand so that you can become the leader?

Happy Learning