

Algorithms and Problem-Solving Lab (15B17CI471)
EVEN 2022
Week -3 (31 Jan – 6 Feb 2023)
Practice Assignment
Topic: Divide and Conquer

Q.1. Cubic integer root x of n is largest number x such that $x^3 \leq n$. Find the value of x given n using divide and conquer approach. Also analyse the complexity.

Q2. Given a sorted array in which all elements appear twice (one after one) and one element appears only once. Find that element in $O(\log n)$ complexity.

Example:

Input: `arr[] = {1, 1, 3, 3, 4, 5, 5, 7, 7, 8, 8}` Output:
4

Input: `arr[] = {1, 1, 3, 3, 4, 4, 5, 5, 7, 7, 8}` Output:
8

Q3. List of points have been given on 2D Plane. Calculate K closest points to the origin (0,0) (Consider euclidean distance to find the distance between two points). Write a code to return the answer in any order. The solution is guaranteed to be unique.

Example 1:

```
Input: points = [[1,3],[-2,2]], K = 1
Output: [[-2,2]]
Explanation:
The distance between (1, 3) and the origin is sqrt(10).
The distance between (-2, 2) and the origin is sqrt(8).
Since sqrt(8) < sqrt(10), (-2, 2) is closer to the origin.
We only want the closest K = 1 points from the origin, so the answer is just
[[-2,2]].
```

Example 2:

```
Input: points = [[3,3],[5,-1],[-2,4]], K = 2
Output: [[3,3],[-2,4]]
(The answer [[-2,4],[3,3]] would also be accepted.)
```

Note:

1. `1 <= K <= points.length <= 10000`
2. `-10000 < points[i][0] < 10000`
3. `-10000 < points[i][1] < 10000`

Q4. Let there be an array of N random elements. We need to sort this array in ascending order. If n is very large (i.e. $N = 1,00,000$) then Quicksort may be considered as the fastest algorithm to sort this array. However, we can further optimize its performance by hybridizing it with insertion sort. Therefore, if n is small (i.e. $N \leq 10$) then we apply insertion sort to the array otherwise Quick Sort is applied. Implement the above discussed hybridized Quick Sort and compare the running time of normal Quick sort and hybridized quick sort. Run each type of sorting 10 times on a random set of inputs and compare the average time returned by these algorithms.

Q5. Consider a sorted array A of n elements. The array A may have repetitive/duplicate elements. For a given target element T , design and implement an efficient algorithm to find T 's first and last occurrence in the array A . Also print the message if an element was not present in the array.

For Example,

Input:

```
arr = [2, 5, 5, 5, 6, 6, 8, 9, 9, 9]
target = 5
```

Output:

```
The first occurrence of element 5 is located at index 1
The last occurrence of element 5 is located at index 3
```

Input:

```
arr = [2, 5, 5, 5, 6, 6, 8, 9, 9, 9]
target = 4
```

Output:

```
Element not found in the array
```