

**AMRITA SCHOOL OF COMPUTING**

**DESIGN AND ANALYSIS OF  
ALGORITHMS  
(23CSE211)**

**Name:** MADDU RAAHITHYA YADAV

**Roll No.:** CH.SC.U4CSE24124

**Class:** BTech (CSE-B)

**School:** Amrita School of Computing,  
Chennai Campus.

**LAB-6**

- 1) Quick Sort using first, last, and random pivot selection methods. Design a menu-driven program that allows the user to choose any method, prints the randomly selected pivot.

Code:

```
#include <stdio.h>
#include <stdlib.h>
int Partition(int arr[], int start, int end, int option) {
    int pivotPos;
    switch (option) {
        case 1:
            pivotPos = start;
            break;
        case 2:
            pivotPos = end;
            break;
        case 3:
            pivotPos = start + rand() % (end - start + 1);
            break;

        default:
            pivotPos = start;
    }
    int temp = arr[start];
    arr[start] = arr[pivotPos];
    arr[pivotPos] = temp;
    int pivot = arr[start];
    int left = start + 1;
    int right = end;
    while (1) {
        while (left <= end && arr[left] <= pivot)
            left++;
        while (arr[right] > pivot)
            right--;
        if (left >= right)
            break;
        temp = arr[left];
        arr[left] = arr[right];
        arr[right] = temp;
    }
}
```

```
temp = arr[start];
arr[start] = arr[right];
arr[right] = temp;
return right;
}
void quickArrange(int arr[], int start, int end, int option) {
    if (start < end) {
        int pos = Partition(arr, start, end, option);
        quickArrange(arr, start, pos - 1, option);
        quickArrange(arr, pos + 1, end, option);
    }
}
int main() {
    int data[] = {157, 110, 147, 122, 111, 149, 151, 141, 123, 112, 117, 133};
    int size = sizeof(data) / sizeof(data[0]);
    int option;
    printf("Choose Pivot Type (Method):\n");
    printf("1. First element\n");
    printf("2. Last element\n");
    printf("3. Random element\n");
    printf("Enter choice: ");
    scanf("%d", &option);
    quickArrange(data, 0, size - 1, option);
    printf("Sorted array:\n");
    for (int k = 0; k < size; k++)
        printf("%d ", data[k]);
    return 0;
}
```

## Output:

```
Choose Pivot Type (Method):
1. First element
2. Last element
3. Random element
Enter choice: 1
Sorted array:
110 111 112 117 122 123 133 141 147 149 151 157
Choose Pivot Type (Method):
1. First element
2. Last element
3. Random element
Enter choice: 2
Sorted array:
110 111 112 117 122 123 133 141 147 149 151 157
Choose Pivot Type (Method):
1. First element
2. Last element
3. Random element
Enter choice: 3
Sorted array:
110 111 112 117 122 123 133 141 147 149 151 157
```

## Space Complexity:

The space complexity of this program is  $O(\log n)$  for best and average cases and  $O(n)$  for worst case.

## Time Complexity:

The time complexity of this program is  $O(n \log n)$  for best and average cases and  $O(n^2)$  for worst case.