



**AMRITA**  
**VISHWA VIDYAPEETHAM**

Second Year B.Tech

(Computer Science and Engineering)

Design and Analysis of Algorithms

**TASK – 5**

Name: Shaik Kolimi Dada Hussain

College Name: Amrita vishwa Vidyapeetham

Roll No : ch.sc.u4cse24144

Department: CSE-B

Academic Year : 2024-2028

# QUICK SORT:

```
#include <stdio.h>
#include <stdlib.h>

void swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}

int partition(int arr[], int low, int high, int pivotChoice) {
    int pivotIndex;
    if (pivotChoice == 1) {
        pivotIndex = low;
    } else if (pivotChoice == 2) {
        pivotIndex = high;
    } else {
        pivotIndex = low + rand() % (high - low + 1);
    }

    printf("Random Pivot Element Chosen: %d\n", arr[pivotIndex]);
    swap(&arr[pivotIndex], &arr[high]);

    int pivot = arr[high];
    int i = low - 1;

    for (int j = low; j <= high - 1; j++) {
        if (arr[j] < pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }

    swap(&arr[i + 1], &arr[high]);
    return i + 1;
}

void quickSort(int arr[], int low, int high, int pivotChoice) {
    if (low < high) {
        int pi = partition(arr, low, high, pivotChoice);
        quickSort(arr, low, pi - 1, pivotChoice);
        quickSort(arr, pi + 1, high, pivotChoice);
    }
}

int main() {
    printf("CH.SC.U4CSE24144\n");

    int n, pivotChoice;
    printf("Enter number of elements: ");
    scanf("%d", &n);

    int *arr = (int *)malloc(n * sizeof(int));

    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Choose Pivot Element:\n");
    printf("1. First Element\n");
    printf("2. Last Element\n");
    printf("3. Random Element\n");
    scanf("%d", &pivotChoice);

    srand(time(NULL));

    quickSort(arr, 0, n - 1, pivotChoice);

    printf("Sorted array:\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    free(arr);
    return 0;
}
```

```
Enter number of elements: 12
Enter 12 elements: 157 110 147 122 111 149 151 141 123 112 117 133
Choose Pivot Element:
1. First Element
2. Last Element
3. Random Element
Choice: 1
Random Pivot Element Chosen: 157
Random Pivot Element Chosen: 133
Random Pivot Element Chosen: 117
Random Pivot Element Chosen: 112
Random Pivot Element Chosen: 111
Random Pivot Element Chosen: 123
Random Pivot Element Chosen: 141
Random Pivot Element Chosen: 147
Random Pivot Element Chosen: 149
Sorted array: 110 111 112 117 122 123 133 141 147 149 151 157
```

#### **TIME AND SPACE COMPLEXITY:**

##### **Time Complexity:**

**Best / Average case:  $O(n \log n)$**

**Worst case:  $O(n^2)$**

##### **Reason:**

- The array is divided into smaller parts repeatedly.
- Worst case happens when the pivot always gives very unbalanced partitions.

##### **Space Complexity:**

**$O(\log n)$  (average)**

**$O(n)$  (worst case)**

##### **Reason:**

- Extra space is used only for recursive function calls (call stack).