

QUICK SORT

Q.2) Sort a given set of N integer elements using the Quick Sort technique and compute its time taken.

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

```
int partition(int A[], int si, int ei)
```

```
{
    int i, j, pivot, temp;
    i = si + 1;
    pivot = A[si];
    j = ei;

    while (i <= j)
    {
        while (A[i] <= pivot)
            i++;
        while (A[j] > pivot)
            j--;

        if (i < j)
        {
            temp = A[i];
            A[i] = A[j];
            A[j] = temp;
        }
    }
}
```

```
temp = A[si];
A[si] = A[j];
A[j] = temp;
return j;
}
```

```
void quickSort(int A[], int si, int ei)
```

```
{
    int mid;
```

```

    if (si < ei)
    {
        mid = partition(A, si, ei);
        quickSort(A, si, mid - 1);
        quickSort(A, mid + 1, ei);
    }
}

int main()
{
    int n;
    int A[10];
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    printf("Enter array elements:\n");
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &A[i]);
    }
    printf("\nthe sorted elements are:\n");
    quickSort(A, 0, n - 1);
    for (int i = 0; i < n; i++)
    {
        printf("%d ", A[i]);
    }
}

```

OUTPUT:

```

Enter the number of elements: 5
Enter array elements:
12 7 5 -2 20

the sorted elements are:
-2 5 7 12 20

```