

Experiment 4

WAP to Perform the Comparative analysis of selection sort and merge sort.

Selection sort

Program:-

```
#include <stdio.h>

#include <time.h>

#include <conio.h>

int smallest(int a[], int k, int n, int pos) {

    int small, j;

    small = a[k];

    for (j = k + 1; j < n - 1; j++) {

        if (small > a[j]) {

            small = a[j];

            pos = j;

        }

    }

    return pos;

}

int main() {

    int i, j, n, a[500000], randNum, temp, pos;

    double time;

    clock_t start, end;

    printf("Enter the total input size:");

    scanf("%d", &n);

    start = clock();

    for (i = 0; i < n; i++) {

        randNum = (rand() % 10000);

        a[i] = randNum;

        printf("%d\t", a[i]);

    }

    for (i = 1; i < n - 1; i++) {
```

```

    pos = smallest(a, i, n, pos);
    temp = a[i];
    a[i] = a[pos];
    a[pos] = temp;
}
printf("\nThe sorted list:\nThe numbers:");
for (i = 0; i < n; i++) {
    printf("%d\t", a[i]);}
end = clock();
time = ((double)(end - start) * 1008) / CLOCKS_PER_SEC;
printf("\nTime=%0lf milliseconds", time);
return 0;
}

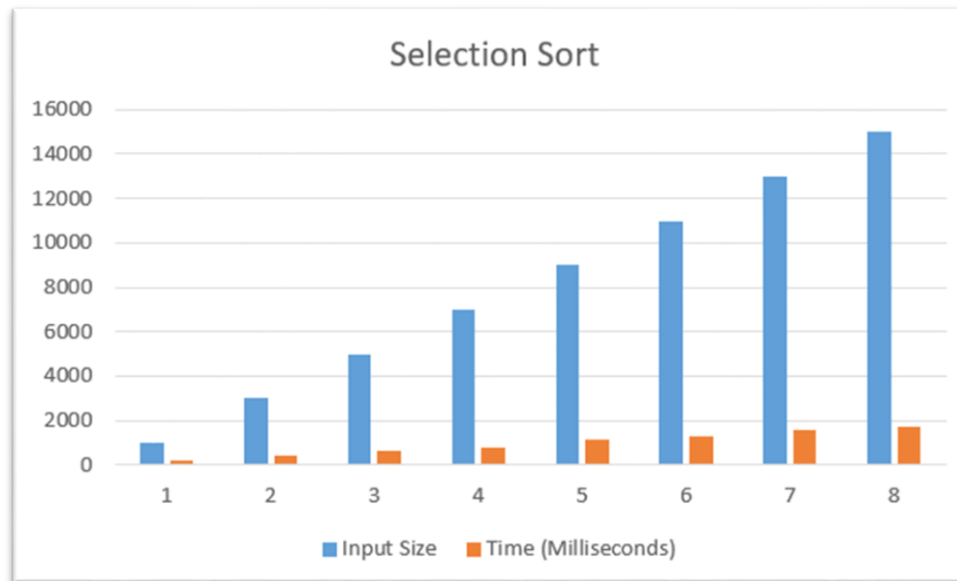
```

Result Analysis and Discussion:

This experiment has been conducted in a 64-bit system with 16 GB RAM and Processor 12th Gen Intel(R) Core (TM) i5-12500H 3.10 GHz. The algorithm is implemented in C programming language in Visual Studio Code 1.83.1 Code Editor. In this experiment the algorithm to sort an array of size “n” using Selection Sort Technique has been implemented and executed for different value of n. During this experiment for different value of n, the time taken by the algorithm has been measured and tabulated as shown in table below.

Input Size	Time (Milliseconds)
1000	169
3000	379
5000	611
7000	798
9000	1145
11000	1304
13000	1537
15000	1717

The graph shown below is the plot of input n and the time in milliseconds taken by the algorithm while running on a system recorded in table above.



Based on the above table and graph it is clearly seen that the size of array n has linear relationship with the time taken by the system to sort an array.

Merge sort

Program:-

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <conio.h>

void mergesort(int a[], int i, int j)
{
    int mid;
    if(i<j)
    {
        mid =(i+j)/2;
        mergesort(a,i,mid);
        mergesort(a,mid+1,j);
        merge(a,i,mid,mid+1,j);
    }
}

void merge(int a[], int i1, int j1, int i2, int j2)
{
    int temp[1000];
    int i,j,k;
    i=i1;
    j=i2,
    k=0;
    while (i<=j1 && j<=j2)
    {
        if(a[i]<a[j])
        {
            temp[k++]=a[i++];
        }
    }
```

```

        else
        {
            temp[k++]=a[j++];
        }
    }
    while (i<=j1)
    {
        temp[k++]=a[i++];
    }
    while (j<=j2)
    {
        temp[k++]=a[j++];
    }
    for (i=i1,j=0;i<=j2;i++,j++)
    {
        a[i]=temp[j];
    }
}

int main()
{
    int a[500000],n,i,randNum;
    double time;
    clock_t start,end;
    printf("Enter the total input size:");
    scanf("%d",&n);
    printf("array elements are:\n");
    start=clock();
    for (i=0;i<n;i++)
    {
        randNum=(rand()%10000);

```

```

        a[i]=randNum;
        printf("%d\t",a[i]);
    }
    mergesort(a,0,n-1);
    printf("\nSorted array is :");
    for (i=0;i<n;i++)
    {
        printf("%d",a[i]);
    }
    end=clock();
    time=((double)(end-start)*1000)/CLOCKS_PER_SEC;
    printf("\nTime=%lf milliseconds",time);
    return 0;
}

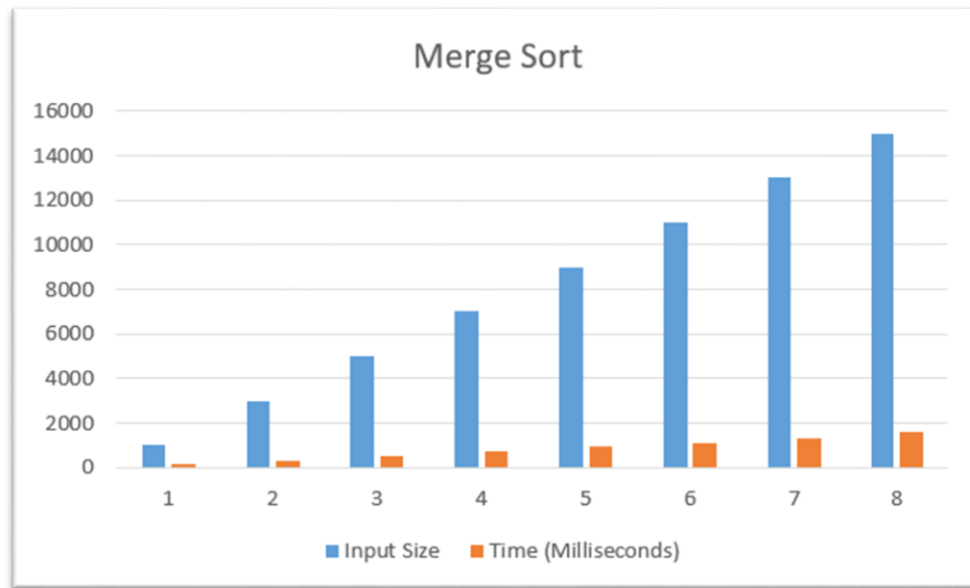
```

Result Analysis and Discussion:

This experiment has been conducted in a 64-bit system with 16 GB RAM and Processor 12th Gen Intel(R) Core (TM) i5-12500H 3.10 GHz. The algorithm is implemented in C programming language in Visual Studio Code 1.83.1 Code Editor. In this experiment the algorithm to sort an array of size “n” using Merge Sort Technique has been implemented and executed for different value of n. During this experiment for different value of n, the time taken by the algorithm has been measured and tabulated as shown in table below.

Input Size	Time (Milliseconds)
1000	130
3000	326
5000	542
7000	740
9000	931
11000	1132
13000	1344
15000	1581

The graph shown below is the plot of input n and the time in milliseconds taken by the algorithm while running on a system recorded in table above.



Based on the above table and graph it is clearly seen that the size of array n has linear relationship with the time taken by the system to sort an array.

Conclusion:

In this experiment, it has been found that the size of input “n” has linear relationship with the time taken by the system to sort the array .So, Merge sort is better than selection sort algorithm to sort an array.