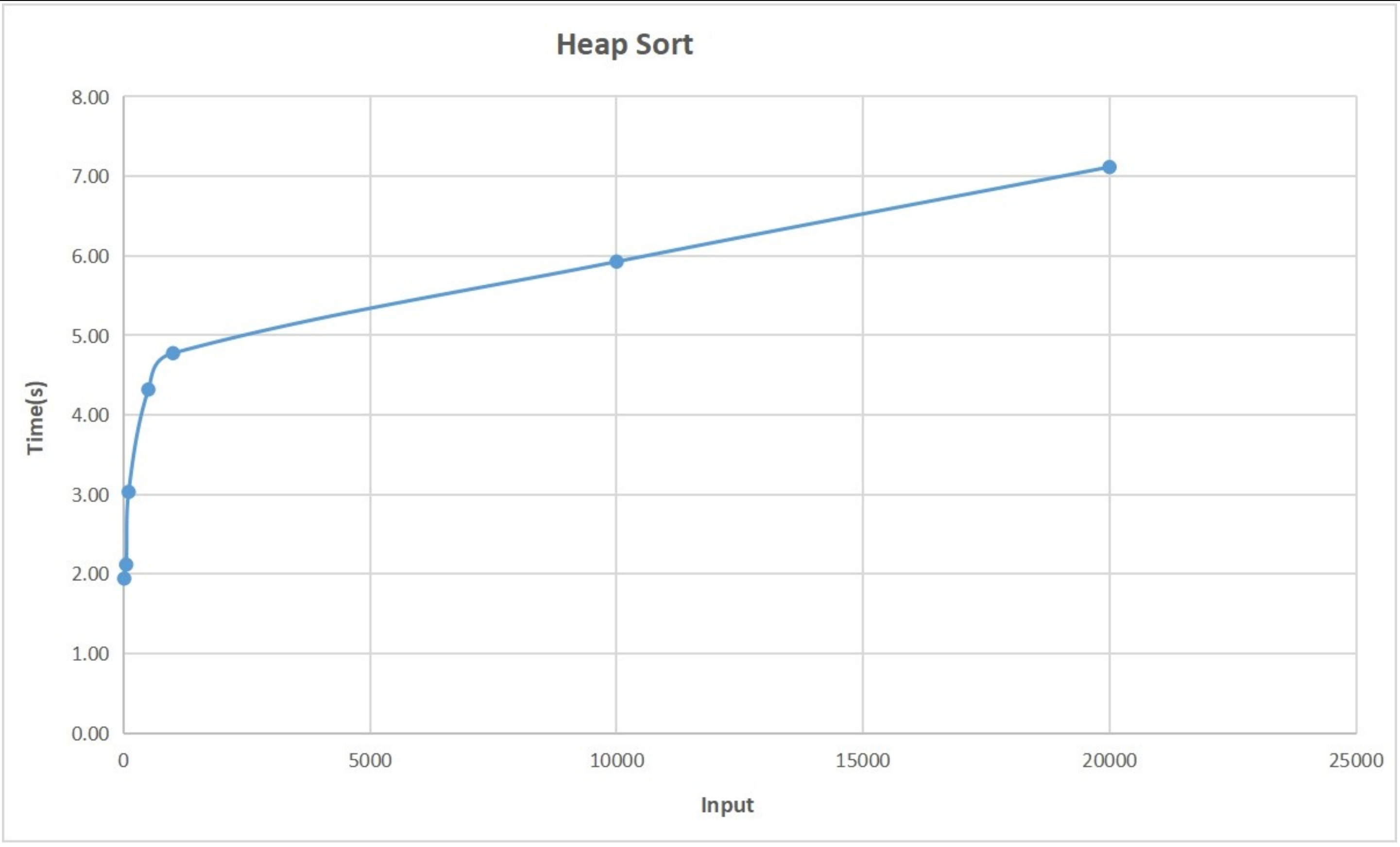
DAA Lab | Ankit Raj | 1906534

Q WAP to Sort a given set of elements using the ~~heap~~ heap sort method and determine the time required to sort the elements. Repeat the experiment for different value of n, and plot the graph of the time taken versus n. The elements can be generated randomly.

Program:
```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void heapify (int arr[], int size, int i){
    int largest = i;
    int left = 2*i+1;
    int right = 2*i + 2;
  if (left < size && arr[left] > arr[largest])
      largest = left;
  if (right < size && arr[right] > arr[largest])
      largest = right;
    if(largest != i){
        int temp = arr[i];
        arr[i] = arr[largest];
        arr[largest] = temp;
       heapify(arr, size, largest);
    }
 }

void ~~hea~~ heapSort(int arr[], int size){
        int i;
  for(i=size/2 -1 ;  i>= 0 ; i--)
         heapify (arr, size, i);

  for (i = size -1 ; i >=0 ; i --){
        int temp= arr[0];
         arr [0] = arr[i];
         arr[i] = temp;
        heapify (arr, i, 0);
     }
  }
```

```
Void main(){
int size;
clock_t start, end;
double total_cputime;
 Start = clock();
 printf("Enter the size:");
  Scanf("%d", &size);
  int arr[size];
  for(int i=0; i<size; i++)
        arr[i]= r and % 10000;

heap Sort(arr, size);
printf(" Array after heap sort:\n");

for(int i=0; i<size; i++)
        printf("%d ", arr[i]);

end=clock();
printf("\n CPU Time Calculation \n");
printf("\n Start time (in ms): %ld", Start);
printf("\n End time (in ms): %ld", end);

total_cputime = ((double)(end - start));
printf("\n Total CPU time (in ms): %f",
                              total_cputime);

total_cputime =((double)(end-Start))/CLOCKS_PER_SE
                                                  ;
printf("\n Total CPU time (in sec): %f",
                                total_cputime);
}
```


Heap Sort
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
Time(s)
0
5000
10000
15000
20000
25000
Input

Q WAP to use divide and conque method to recursively implement and to find the maximum and minimum in a given list of n elements.

<u>program</u>:
```
# include <stdio.h>
int max, min;
int a[100];
void maxmin (int i, int j){
    int max1, min1, mid;
    if (i==j){
        max = min = a[i];
    }
    else {
        if (i== j-1){
            if (a[i] < a[j] ){
                max = a[j];
                min = a[i];
            }
            else {
                max=a[i];
                min= a[j];
            }
        }
```

Scanned by TapScanner

```
else {
      mid = (i+j)/2;
    maxmin (i, mid);
      max1 = max;
      min1 = min;

      max min (mid+1, j);
      if (max < max1)
            max = max1;
       if (min > min1)
             min = min1;
      }
    }
  }

int main (){
     int i, n;
    printf ("Enter size of array:");
    Scanf ("%d", &n);
   printf(" Enter elements:\n");
   for (i=1; i<=n; i++)
            Scanf ("%d", &a[i]);
    max = a[0];
    min = a[0];

maxmin (1, n)
printf("Minimum element in the array: %d\n",
                                        min);
printf("maximum element in the array: %d\n",
                                        max);
 return 0;
}
```

INPUT/OUTPUT :-

```
 Enter size of array : 10
 Enter elements :
 22  13  -5  -8  15  60  17 31 7 14

 Minimum elements in the array : -8
 Maximum element in the array : 60.
```

Scanned by TapScanner