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\t 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_cpu time);

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