Week -06 >

Quick Sort time complexity

PAGE NO :
DATE :

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

Void quick-sort (int a[],int low, int high);
int partition (int a[],int low, int high);

int main() {
    int a[15000], n, i, j, ch, temp;
    clock-t, start, end;

    while (1) {
        printf("\n1: for manual entry of N value \n");
        printf("2: To display time taken for random no \n");
        printf("3: Exit \n");
        Printf(" Enter your choice: ");
        scanf("%d", &ch);

        switch (ch) {
            case 1:
                Printf ("\n Enter number of elements: ");
                scanf("%d", &n);
                printf("Enter array elements: ");
                for (i=0; i<n; i++) {
                    scanf ("%d", &a[i]);
                }
                start = clock ();
                Quick_sort (a, 0, n-1);
                end = clock ();
                printf("\n sorted array is: ");
                for (i=0; i<n; i++) {
                    Printf("%d \t", a[i]);
                }
```

PAGE NO :
DATE :

```
// Start = clock();
// quick_sort(a, 0, n-1);
// end = clock();
// printf("\n sorted array is: ");

printf("\n Time taken to sort %d number is %f
                secs", n, (((double)(end - start))/
                                        clocks PER SEC));

  break;

Case 2:
    n = 5000;
    while(n <= 5000000) {
      for(i=0; i<n; i++) {
         a[i] = n-i;
      }
       start = clock();
       quick_sort(a, 0, n-1);
       for(j=0; j<50000; j++) {
            temp = 38/600;
       }
       end = clock();
       printf("\nTime taken sort %d number is
       %f secs", n, (((double)(end-start))/clock_sec));
       n = n+1000;
    }
    break;
  case 3:
    exit(0);
 }
 getchar();
}
```

PAGE NO :
DATE :

```
void quick_sort(int a[], int low, int high) {
    if (low < high) {
        int pi = partition(a, low, high);
          quick_sort (a, low, pi-1);
          quick_sort(a, pi+1, high);
    }
}
int partition (int a[], int low, int high) {
    int pivot = a[high];
    int i = (low - 1);
    for(int j = low; j<= high - 1; j++) {
        if(a[j] < pivot) {
            i++;
            int temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
    int temp = a[i+1];
    a[i+1] = a[high];
    a[high] = temp;
    return (i+1);
}
```

Out Put :-

```
1: For manual entry of N value.
2: To disply time taken sorting num of ele
3: To Exit.
Enter your choice: 1
Enter number of elements: 10.
Enter array elements:
12  24  33  5  7  19  10  11  22  30
```

PAGE NO :
DATE :

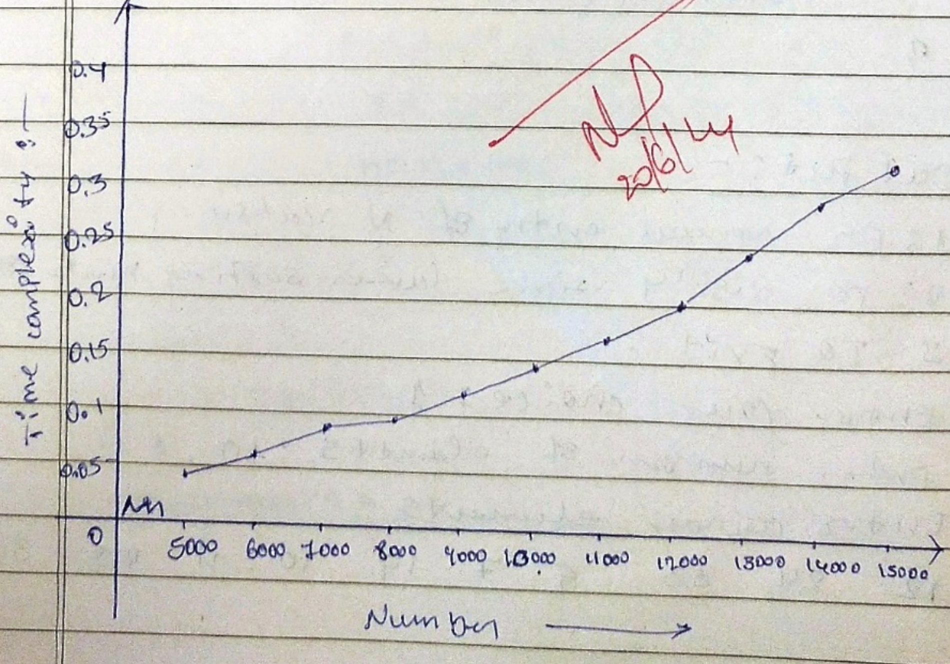
```
Sorted array is :
5   7   10   11   12   19   22   24   80   83
Time taken to sort 10 number is 0.000001 secs.

Enter your choice : 2

Time taken to sort 5000 number is 0.037644 Secs
   -"-            6000   -"-     0.053973 Secs
   -"-            7000   -"-     0.083635 Secs
   -"-            8000   -"-     0.095201 Secs
   -"-            9000   -"-     0.119518 Secs
   -"-           10000   -"-     0.143924 Secs
   -"-           11000   -"-     0.173608 Secs
   -"-           12000   -"-     0.206768 Secs
   -"-           13000   -"-     0.2419991 Secs
   -"-           14000   -"-     0.281033 Secs
   -"-           15000   -"-     0.322459 Secs

Enter your choice : 3
    Exited successfully !
```

Graph :-



[illegible] 20/6/24