Practical-6

(1) Generate large number of elements randomly and sort all the elements in ascending order using Quick sort. Analyse the time complexity for best, average and worst case.

Code:

```
#include<stdio.h>
#include<time.h>
#include<sys/time.h>
int get_pivot(int a[],int first,int last)
{
   int pivot,p,q,temp;
   pivot=a[first];
   p=first;
   q=last;
   while(p<=q)
   {
      while(a[p]<=pivot)
      {
         p++;
      }
      while(a[q]>pivot)
      {
         q--;
      }
      if(p<q)
      {
         temp=a[p];
         a[p]=a[q];
         a[q]=temp;
      }
   }
   temp=a[first];
   a[first]=a[q];
   a[q]=temp;
   return q;
```

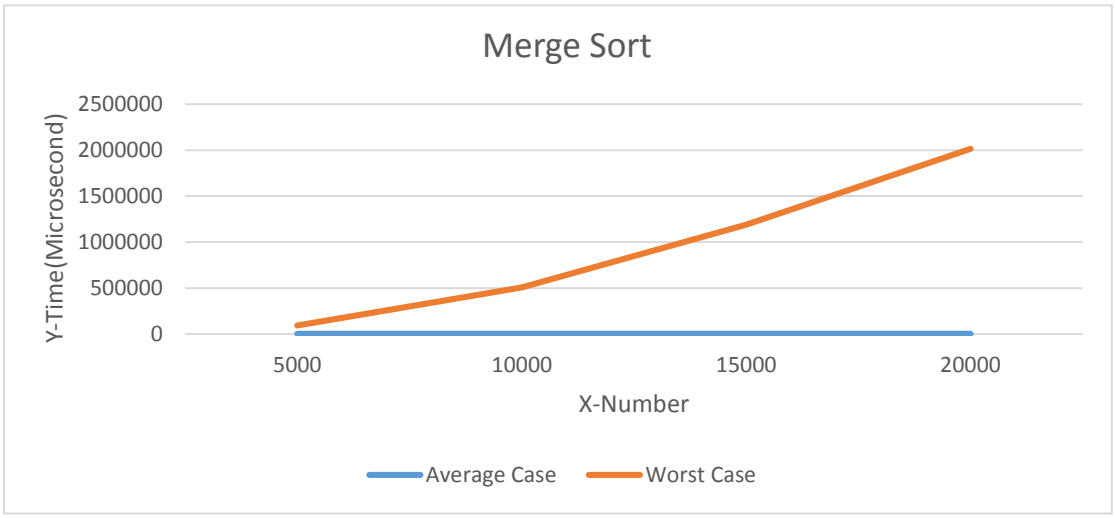
```
}
void quicksort(int a[],int first,int last)
{

   int m;
   if(first<last)
   {
     m=get_pivot(a,first,last);
     quicksort(a,first,m-1);
     quicksort(a,m+1,last);
   }
}
void checkfor(int a[],int n,int i)
{
   int t2,t1;
   struct timeval tv;
   struct timezone tz;

   //Average case

   for(i=0;i<n;i++)
   {
      a[i]=rand()%n;
   }
   gettimeofday(&tv,&tz);
   t1=((tv.tv_sec*1000000)+(tv.tv_usec));
   quicksort(a,0,n-1);
   gettimeofday(&tv,&tz);
   t2=((tv.tv_sec*1000000)+(tv.tv_usec));
   printf("\n %d \t\t %d ",n,(t2-t1));


   //worst case

   for(i=0;i<n;i++)
   {
      a[i]=n-i;
   }
```

```
   gettimeofday(&tv,&tz);
   t1=((tv.tv_sec*1000000)+(tv.tv_usec));
   quicksort(a,0,n-1);
   gettimeofday(&tv,&tz);
   t2=((tv.tv_sec*1000000)+(tv.tv_usec));
   printf("\t\t%d",(t2-t1));
}
void main()
{
  int a[20000],n,i;
  printf("\n Value     \tAverage case\tWorst case\n");
  n=5000;
  checkfor(a,n,i);
  n=10000;
  checkfor(a,n,i);
  n=15000;
  checkfor(a,n,i);
  n=20000;
  checkfor(a,n,i);
}
```

Output-Table:

Value	Average Case	Worst Case
5000	888	95638
10000	1532	508379
15000	3611	23191061
20000	3077	2015260

Graph:


Merge Sort
Y-Time(Microsecond)
2500000
2000000
1500000
1000000
500000
0
5000
10000
15000
20000
X-Number
Average Case
Worst Case