6/6/24. Quick Sort

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

void split(int [], int, int);
int partition(int [], int, int);

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

    while(1){
        printf("1:For manual entry of N value
                and array elements");
        printf("\n2:To display time taken for
        sorting number of elements N in the
        range 500 to 14500");
        printf("\n3 :To exit");
        printf("\n Enter your choice:");
        scanf("%d", &ch);

       switch(ch){
          case 1:
          printf("Enter the number of elements:");
          scanf("%d", &n);
          printf("Enter array elements:");
          for(i=0; i<n; i++){
                scanf("%d, & a[i]);
          }
          start = clock();
          split(a, 0, n-1);
          end = clock();
printf ("sorted array is:");
for (i =0; i<n; i++){
        printf ("%d\t", a[i]);
};
prinf ("Time taken to sort %d numbers is
 %f Secs \n", n, ((double) (end-start)) /
                clocks per sec);
break;
case 2:
     n=500;
     while (.n<=14500){
         for (i=0; i<n; i++){
               a[i]=n-i;
         }
         start=clock();
         split(a, 0, n-1);
         for (j=0; j<50000000; j++)
         { temp=38/600; }
         end=clock();
         printf ("Time taken to sort %d", n,
         ((double) (end-start)) /clocks per sec)
         n+=1000;
     } break;

case 3:
     exit(0);

default:
     printf ("\n Invalid choice! Please try
               again \n");
   }
 }
return 0;
}
```

6/6/24

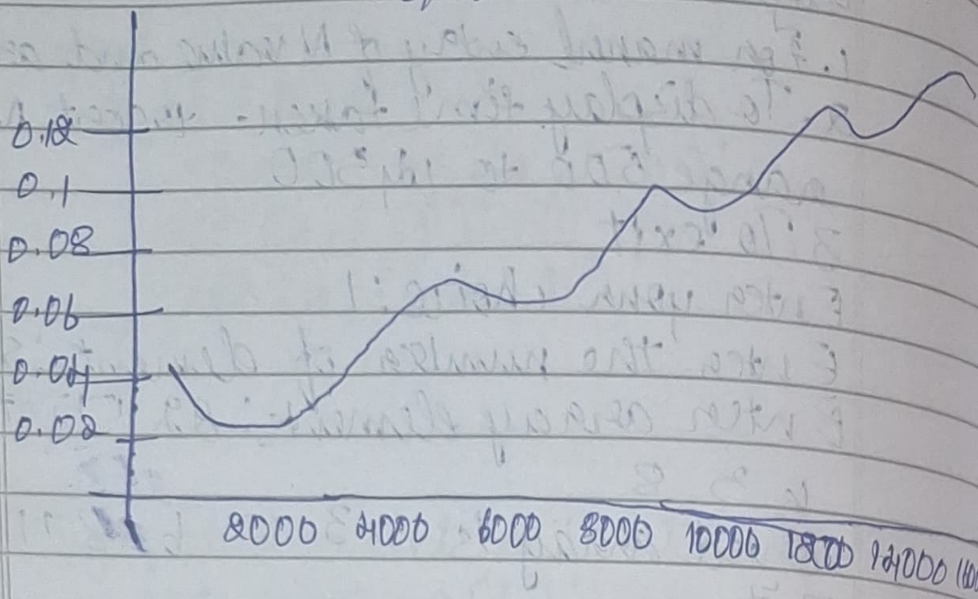
```
void split(int a[], int l, int h){
  int j;
  if (l<h) {
     j=partition (a,l,h);
     split (a,l,j);
     split (a,j+1,h);
   }
}

int partition(int a[], int l, int h){
     int pivot=a[l];
     int i=l+1;
     int j=h;
     int temp,t;
     do
     {
        while(i<=h && pivot>=a[i])
        { i++; }
       while (pivot< a[j])
       { j--; }
       if (i<j)
       {
          temp=a[i];
          a[i]=a[j];
          a[j]=temp;
       }
    }while(i<j);
    t=a[l];
    a[l]=a[j];
    a[j]=t;
    return j;
}
```

Output

```
1. For manul entry of N value and array elements
2. To display time taken to sort N in the
 range 500 to 14500
3. To exit
Enter your choice: 1
Enter the number of elements: 8
Enter array elements: 22 55 77 11 5
 6 3 8
sorted array is 3 5 6 8 11 22 55
77
Time taken to sort 8 numbers is 0 secs
1. For manul entry of N value and
array elements
2. To display time taken to sort N in
the range 500 to 14500
3. To exit
Enter your choice: 2
Time taken to sort 500 numbers: 0.031 secs
Time taken to sort 1500 numbers: 0.016 secs
Time taken to sort 2500 numbers: 0.016 secs
Time taken to sort 3500 numbers: 0.015 secs
Time taken to sort 4500 numbers: 0.031 secs
Time taken to sort 5500 numbers: 0.047 secs
Time taken to sort 6500 numbers: 0.047 secs
Time taken to sort 7500 numbers: 0.047 secs
Time taken to sort 8500 numbers: 0.063 secs
Time taken to sort 9500 numbers: 0.062 secs
Time taken to sort 10500 numbers: 0.078 secs
Time taken to sort 11500 numbers: 0.094 secs
Time taken sort 12500 numbers: 0.094 secs
Time taken sort 13500 numbers: 0.109 secs
Time taken sort 14500 numbers: 0.11 secs
```

## Gaaph

Quick Sort


0.12
0.1
0.08
0.06
0.04
0.02
2000
4000
6000
8000
10000
1800
14000
160