6/6/24

Quick sort

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

void swap (int *a, int *b)
{
        int temp = *a;
        *a = *b;
        *b = temp;
}
int partition (int A[], int low, int high)
{
        int pivot = A[low];
        int i = low;
        int j = high + 1;

        do {
            do {
                    i++;
                } while (A[i] < pivot);
            do {
                    j--;
                } while (A[j] > pivot);
            if (i < j)
                    swap (&A[i], &A[j]);
        } while (i < j);

        swap (&A[low], &A[j]);
        return j;
}
```

```
void quicksort (int A[], int low, int high)
{
        if (low < high)
        {
                int split = partition (A, low, high);
                quicksort (A, low, split -1);
                quicksort (A, split +1, high);
        }
}

int main() {
        int n, ch;
        clock_t start, end;
        while(1) {
            printf("\n1 For manual entry of N and
                              array elements");
            printf("\n2 : To display time taken for
                              sorting no. of elements N
                              in range 500 to 14500")
            printf("\n3: exit");
            printf("\n Enter your choice");
            scanf("%d", &ch);

            switch (ch)
            {
                    case 1:
                            printf("Enter no. of elements\n");
                            scanf("%d", &n);
                            int *A = (int*) malloc (n* sizeof(int))
                            if (A == NULL) {
                                printf("Memory allocation failed")
                                exit(1)
                            }
```

13/6/24

```
printf (" Enter array elements \n");
for (int i=0; i<n; i++)
{
      scanf ("%d", &A[i]);
}

start = clock();
quicksort (A, 0, n-1);
end = clock();
printf ("\n Sorted array is :");
for(int i=0; i<n; i++)
{
      printf("%d\t", A[i]);
}

free (A);
break;


case 2:
      n = 500
      while (n <= 14500)
      {
            int *A = (int *) malloc (n* sizeof (int
            if (A == NULL) {
                  printf (" Memory allocation fai
                  exit(1);
            }
            for(int i=0; i<n; i++)
            {
                  A[i] = n-i;
            }
            start = clock();
```

~~printf~~

```
    quicksort (A, 0, n-1);
    for(int j=0; j<500000; j++)
    {
        int temp = 38/600;
    }
    end = clock();
    printf("Time taken to sort %d numbers is
       %f seconds\n", n, ((double)(end-start))/
                                CLOCKS_PER_SEC);
    free(A);
    n += 1000;
   }
   break;

  case 3:
       exit(0);
       break;

  default:
       printf("Invalid choice \n");

  }
 }
 return 0;
}
```

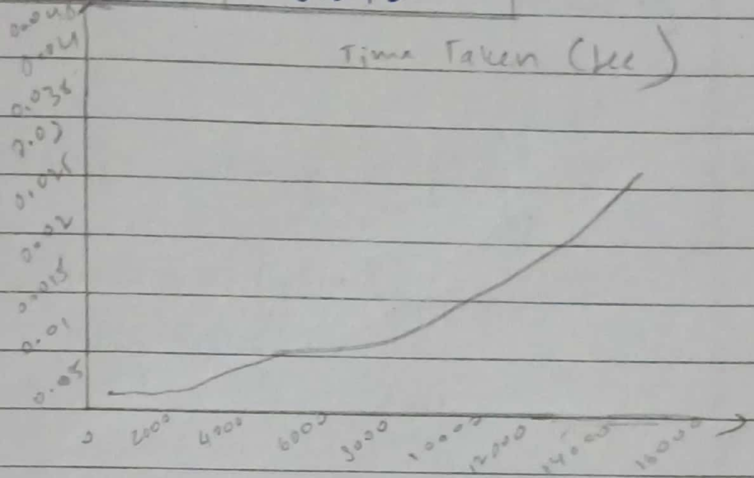
→ output

```
1: For manual entry of N value and array
   elements

2: To display time taken for sorting
   no. of elements N in the range
   500 to 14500

3: To exit

Enter your choice : 1

Enter no. of elements : 5
Enter array elements:
5
2
8
9
1

Sorted array is : 1 2 5 8 9
Time taken to sort 5 numbers is 0 sec
```

Graph

N	Time taken (secs)
500	0.001
1500	0.001
2500	0.002
3500	0.003
4500	0.004

5500	0.007	
6500	0.009	
7500	0.011	
8500	0.015	
9500	0.019	
10500	0.023	
11500	0.027	
12500	0.031	
13500	0.037	
14500	0.042	


Time Taken (Sec)