

6/6/24

⇒ Quicksort : - LAB ⑥

```
#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 + 1;
    int j = high;
    while (i <= j)
    {
        while (i <= j && a[i] <= pivot)
            i++;
        while (i <= j && a[j] >= pivot)
            j--;
        if (i < j)
            swap(&a[i], &a[j]);
    }
    swap(&a[low], &a[j]);
    return j;
}

void quicksort(int a[], int low, int high)
{
    int point;
    if (low < high)
        partition(a, low, high);
}
```

```

    {
        point = partition (a, low, high);
        quicksort (a, low, point - 1);
        quicksort (a, point + 1, high);
    }
}

void main()
{
    int a[50000], n, i, j, ch, temp;
    clock_t start, end;
    while(1)
    {
        printf("1. For manual entry of N. value & array elements");
        printf("2. To display time taken for sorting no. of elements N in the range 500 to 14500");
        printf("3. To exit");
        printf("Enter your choice:");
        scanf("%d", &ch);
        switch(ch)
        {
            case 1:
                printf("Enter no. of elements:");
                scanf("%d", &n);
                printf("Enter array elements:");
                for(i=0; i<n; i++)
                {
                    scanf("%d", &a[i]);
                }
                start = clock();
                quicksort(a, 0, n-1);
                end = clock();
                printf("Sorted array is:\n");
        }
    }
}

```

```

for(i=0; i<n; i++)
{
    printf("%d\t", a[i]);
}
printf("Time taken to sort y.d no.s is %f secs", 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; // for boundary test i.e. "0"
    }
    quicksort(a, 0, n-1); // for testing
    for(j=0; j<500000; j++)
    {
        temp = 38/600;
    }
    end = clock();
    printf("Time taken to sort y.d. no.s is %.6f secs",
          ((n, ((double)(end-start)) / CLOCKS_PER_SEC)));
    n = n + 1000;
}
break;

Case 3:
exit(0);
}
getchar(); // for testing
}

```

Output:

1. For manual entry of N-values & 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 the no. of elements: 8

Enter array elements: 5 3 1 9 8 2 4 7

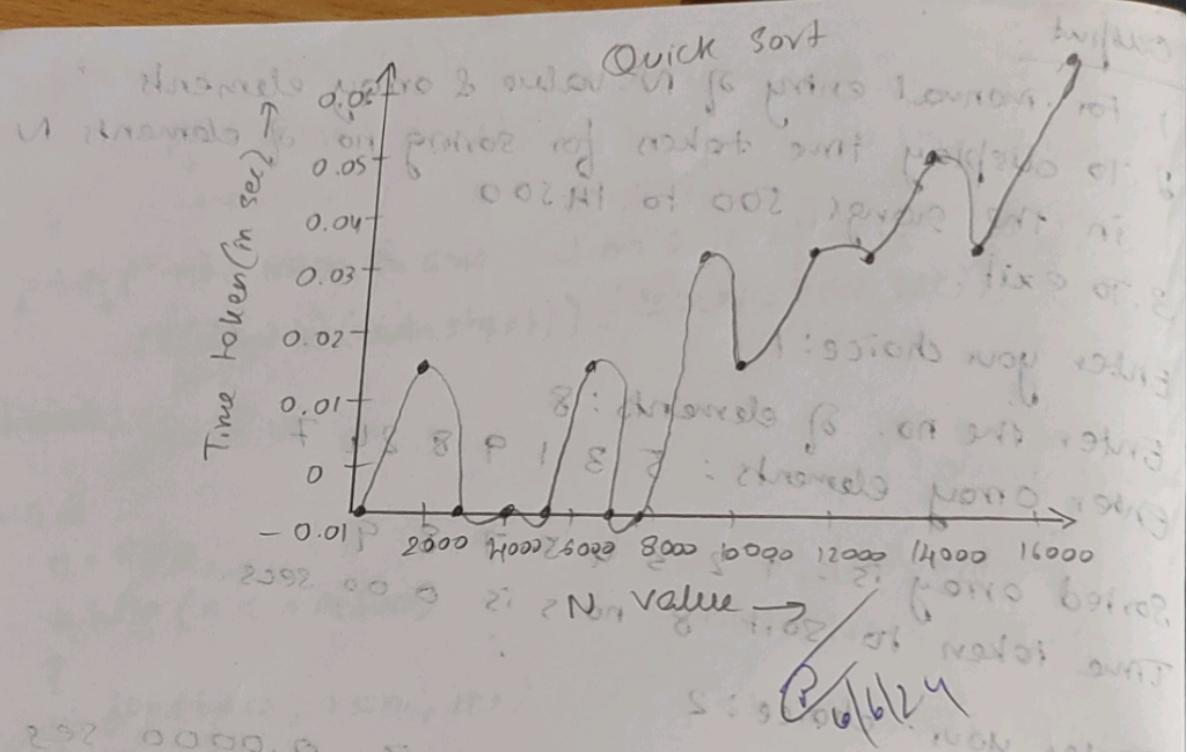
Sorted array is: 1 2 3 4 5 7 8 9

Time taken to sort 8 nos is 0.00 secs.

Enter your choice: 2

Time taken to sort	500 nos	is	0.0000	secs
" " "	1500	"	0.016	secs
" " "	2500	"	0.000	secs
" " "	3500	"	0.000	secs
" " "	4500	"	0.000	secs
" " "	5500	"	0.015	secs
" " "	6500	"	0.000	secs
" " "	7500	"	0.000	secs
" " "	8500	"	0.031	secs
" " "	9500	"	0.016	secs
" " "	10500	"	0.031	secs
" " "	11500	"	0.031	secs
" " "	12500	"	0.047	secs
" " "	13500	"	0.032	secs
" " "	14500	"	0.062	secs

~~Enter your choice: 3~~



2392	0000.0	8	2001	002	1020	0028	0021	0020	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z
2392	0000.0	"	2002	002	1021	0029	0022	0021	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z	
2392	0000.0	"	2003	002	1022	002A	0021	0020	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z	
2392	0000.0	"	2004	002	1023	002B	0021	0020	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z		
2392	0000.0	"	2005	002	1024	002C	0021	0020	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z			
2392	0000.0	"	2006	002	1025	002D	0021	0020	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z				
2392	0000.0	"	2007	002	1026	002E	0021	0020	0027	0028	0029	002A	002B	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z					
2392	0000.0	"	2008	002	1027	002F	0021	0020	0028	0029	002A	002B	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z						
2392	0000.0	"	2009	002	1028	002G	0021	0020	0029	002A	002B	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z							
2392	0000.0	"	2010	002	1029	002H	0021	0020	002A	002B	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z								
2392	0000.0	"	2011	002	102A	002I	0021	0020	002B	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z									
2392	0000.0	"	2012	002	102B	002J	0021	0020	002C	002D	002E	002F	002G	002H	002I	002J	002K	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z										
2392	0000.0	"	2013	002	102C	002K	0021	0020	002D	002E	002F	002G	002H	002I	002J	002L	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z												
2392	0000.0	"	2014	002	102D	002L	0021	0020	002E	002F	002G	002H	002I	002J	002K	002M	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z													
2392	0000.0	"	2015	002	102E	002M	0021	0020	002F	002G	002H	002I	002J	002K	002L	002N	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z														
2392	0000.0	"	2016	002	102F	002N	0021	0020	002G	002H	002I	002J	002K	002L	002M	002O	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z															
2392	0000.0	"	2017	002	102G	002O	0021	0020	002H	002I	002J	002K	002L	002M	002N	002P	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z																
2392	0000.0	"	2018	002	102H	002P	0021	0020	002I	002J	002K	002L	002M	002N	002O	002Q	002R	002S	002T	002U	002V	002W	002X	002Y	002Z																	
2392	0000.0	"	2019	002	102I	002Q	0021	0020	002J	002K	002L	002M	002N	002O	002P	002R	002S	002T	002U	002V	002W	002X	002Y	002Z																		
2392	0000.0	"	2020	002	102J	002R	0021	0020	002K	002L	002M	002N	002O	002P	002Q	002S	002T	002U	002V	002W	002X	002Y	002Z																			
2392	0000.0	"	2021	002	102K	002S	0021	0020	002L	002M	002N	002O	002P	002Q	002R	002T	002U	002V	002W	002X	002Y	002Z																				
2392	0000.0	"	2022	002	102L	002T	0021	0020	002M	002N	002O	002P	002Q	002R	002S	002U	002V	002W	002X	002Y	002Z																					
2392	0000.0	"	2023	002	102M	002U	0021	0020	002N	002O	002P	002Q	002R	002S	002T	002V	002W	002X	002Y	002Z																						
2392	0000.0	"	2024	002	102N	002V	0021	0020	002O	002P	002Q	002R	002S	002T	002U	002W	002X	002Y	002Z																							
2392	0000.0	"	2025	002	102O	002W	0021	0020	002P	002Q	002R	002S	002T	002U	002V	002X	002Y	002Z																								
2392	0000.0	"	2026	002	102P	002X	0021	0020	002Q	002R	002S	002T	002U	002V	002W	002Y	002Z																									
2392	0000.0	"	2027	002	102Q	002Y	0021	0020	002R	002S	002T	002U	002V	002W	002X	002Z																										
2392	0000.0	"	2028	002	102R	002Z	0021	0020	002S	002T	002U	002V	002W	002X	002Y																											

E: sort by value