6/6/24

LAB-5 Qu

```
void swap (int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
int partition (int a[], int low, int high)

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

void quicksort (int arr[], int low, int high)
{
    if (low < high)
        int pi = partition (arr, low, high)

        quicksort (asr, low, pi-1);
        quicksort (arr, pi+1, high);
    }
}
```

```
void main ()<
    int a [150000] , n, i, j, ch, temp;
    clock_t start, end;

    while (1) <
        pf ("\n 1: Enter N value & elements ");
        pf ("\n 2: To display time taken for sorting
            number of element N in range 500 to
            140000");
        pf ("\n 3: To exit ");
        pf ("\n Enter choice : ");
        scanf ("%d", &ch);

    switch (ch) <
        case 1:
            pf ("\n Enter the no. of element ");
            scanf ("%d", &n);
            printf ("Enter elements :");
            for (i=0; i<n; i++) <
                scanf ("%d", &a[i]);
            }
            start = clock();
            quicksort (a, 0, n-1);
            end = clock();
            pf ("\n sorted array : ");
            for (i=0; i<n; i++) <
                pf ("%d \t", a[i]);
            }
        pf ("\n Time taken to sort %d numbers is
            %f secs", n, ((double)(end-start))/
                Clocks_per_sec));
        break;
```

```
case 2:
    n=500;
    while (n<=14000){
        for (i=0; i<n; i++){
            a[i] = n-i;
        }
        start = clock();
        quicksort (a, 0, n-1);
        for (j=0; j<500000; j++){
            temp = 0.38 / 600;
        }
        end = clock();
        printf ("\n Time taken to sort %d
                numbers is %.f secs", n,
        ((double) (end-start)) / Clocks_per_sec));
        n=n+1000;
    }
    break;

case 3:
        exit (0);
    }
    getchar();
}
```

6/6/24

Output:

Enter no. of elements: 6
Enter array elements: 38 4 5 6 20 50
Sorted array is: 4 5 6 20 38 50

Page

Graph: