Name : Saikat Sheet

University Roll : 18700120024

Department : CSE

Sec : A

Assignment 2.1

#include <stdio.h>

int array[] = {10, 20, 45, 50, 56, 64, 78, 89, 91, 100};

int dac\_count;

int maxmin1()

{

int count = 0;

int max = array[0], min = array[0];

for(int i = 0; i<10; i++)

{

count++;

if(array[i] > max) max = array[i];

count++;

if(array[i] < min) min = array[i];

}

return count;

}

int maxmin2()

{

int count = 0;

int max = array[0], min = array[0];

for(int i = 0; i<10; i++)

{

count++;

if(array[i] > max) max = array[i];

else

{

count++;

if(array[i] < min) min = array[i];

}

}

return count;

}

int maxmin\_dac(int low, int high, int max, int min, int count)

{

dac\_count++;

if(array[low] == array[high])

{

max = array[low];

}

else

{

dac\_count++;

if(low == 8)

{

dac\_count++;

if (array[low] < array[high])

{

min = array[low];

max = array[high];

}

else

{

max = array[low];

min = array[high];

}

}

else

{

int mid = (low+high)/2;

int max1 = 0, min1 = 0;

maxmin\_dac(low, mid, max, min, count);

maxmin\_dac(mid+1, high, max1, min1, count);

dac\_count++;

max = (max1>max)? max1 : max;

dac\_count++;

min = (min1<min)? min1 : min;

}

}

return dac\_count;

}

int main()

{

int n;

n = maxmin1();

printf("No of comparisons for maxmin1: %d\n", n);

n = maxmin2();

printf("No of comparisons for maxmin2: %d\n", n);

n = maxmin\_dac(0, 9, 10, 10, 0);

printf("No of comparisons for maxmin\_dac: %d\n", n);

return 0;

}

Ouput:



Assignment 2.2:

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

int \*a;

int partition(int p, int r)

{

int x = a[r];

int i = (p-1);

for(int j = p; j <= (r-1); j++)

{

if(a[j] < x)

{

i++;

int temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

int temp = a[(i+1)];

a[(i+1)] = a[r];

a[r] = temp;

return (i+1);

}

void quick\_sort(int p, int r)

{

if(p < r)

{

int q = partition(p, r);

quick\_sort(p, (q-1));

quick\_sort((q+1) , r);

}

}

int main()

{

clock\_t start1 , end1, total\_1;

printf("For 500 input test case :\n");

int n1 = 500;

a = (int\*)malloc(n1 \* sizeof(int));

int l = 1,u = 500;

for(int i=0; i<500; i++)

{

int num = (rand()%(u-l+1)+l);

a[i] = num;

}

start1 = clock();

quick\_sort(0, 499);

end1 = clock();

for(int i = 0; i <= 9; i++)

{

printf("%d " ,a[i]);

}

printf("\n");

total\_1 = (double)(end1 - start1) ;

printf("Total time taken by CPU: %ld\n\n\n",total\_1);

printf("For 1000 input test case :\n");

int n2 = 1000;

a = (int\*)malloc(n2 \* sizeof(int));

int l1 = 1, u1 = 1000;

for(int i=0; i<1000; i++)

{

int num=(rand()%(u1-l1+1)+l1);

a[i]=num;

}

start1 = clock();

quick\_sort(0 , 999);

end1 = clock();

for(int i = 0; i <= 9; i++ )

{

printf("%d " ,a[i]);

}

printf("\n");

total\_1 = (double)(end1 - start1) ;

printf("Total time taken by CPU: %ld\n\n\n", total\_1);

printf("For 5000 input test case :\n");

int n3 = 5000;

a = (int\*)malloc(n3 \* sizeof(int));

int l2 = 1,u2 = 5000;

for(int i=0; i<5000; i++)

{

int num=(rand()%(u2-l2+1)+l2);

a[i]=num;

}

start1 = clock();

quick\_sort(0 , 4999);

end1 = clock();

for(int i = 0; i <= 9; i++ )

{

printf("%d " ,a[i]);

}

printf("\n");

total\_1 = (double)(end1 - start1) ;

printf("Total time taken by CPU: %ld\n\n\n", total\_1);

printf("For 10000 input test case :\n");

int n4 = 10000;

a = (int\*)malloc(n4 \* sizeof(int));

int l3 = 1,u3 = 10000;

for(int i=0; i<10000; i++)

{

int num=(rand()%(u3-l3+1)+l3);

a[i]=num;

}

start1 = clock();

quick\_sort(0 , 9999);

end1 = clock();

for(int i = 0; i <= 9; i++ )

{

printf("%d " ,a[i]);

}

printf("\n");

total\_1 = (double)(end1 - start1) ;

printf("Total time taken by CPU: %ld\n", total\_1);

}

Output:

