**3.QUICK SORT**

#include<iostream>

#include<cstdlib>

#include<time.h>

#include <cmath>

#include <fstream>

using namespace std;

int no\_comp=0;

int getRandom(int p,int r){

srand(time(0)); // 4 7

return p+(rand()% (r-p) ); // 4 ---> 0 , 1, 2 , 3 + 4

}

int partition(int \*arr,int p, int r){ // crux of quick sort

int y = getRandom(p,r);

swap(arr[y],arr[r]);

int x= arr[r];

int i= p-1;

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

if (++no\_comp || arr[j]<=x){

i=i+1;

swap(arr[i],arr[j]) ;

}

}

swap(arr[i+1],arr[r]);

return i+1;

}

void quicksort(int \*arr,int p, int r){

if(p>=r)

return;

int q=partition(arr,p,r);

quicksort(arr,p,q-1);

quicksort(arr,q+1,r);

}

int\* getRandomEveryTime(int range,int s){ //gives the sizes of array

int \*arr=new int[s];

int flag=0;

for(int i=0;i<s;){

flag=0;

int x=rand()% range+30; //0 - 970 0-->30

for(int j=0;j<i;j++){

if(arr[j]==x){

flag=1;

break;

}

}

if(flag==0){

arr[i]=x;

i++;

}

}

return arr;

}

void print\_arr(int \*arr, int s){

for(int i=0;i<s;i++){

cout<<arr[i]<<" , ";

}

}

int \*getRandomx(int range,int s){ //gives the sizes of array

int \*arr=new int[s];

for(int i=0;i<s ;i++){

arr[i]=rand()% range+30;

}

return arr;

}

int main(){

ofstream nlogn("nlogn\_quick.txt");

ofstream nsq("nsq\_quick.txt");

ofstream nsize("n\_quick.txt");

ofstream comp("comp\_quick.txt");

int s =100; // 100 different sizes of inputs

int \*arr=getRandomEveryTime(970,s); // 30 - 1000

for(int i=0 ;i<s;i++){

int siz = arr[i];

int \*arrx = getRandomx(100,siz);

quicksort(arrx,0,siz-1);

double n = siz;

cout<<"NO OF COMP. = "<<no\_comp<<" for N = " <<siz<<endl;

no\_comp =0;

}

return 0;

}

