/\*JATINDER DHAWAN-14103025\*/

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<time.h>

#include<stdlib.h>

void quick\_sort(long int \*x,int first,int last){

long int pivot,j,temp,i;

if(first<last)

{

pivot=first;

i=first;

j=last;

while(i<j){

while(x[i]<=x[pivot]&&i<last)

i++;

while(x[j]>x[pivot])

j--;

if(i<j){

temp=x[i];

x[i]=x[j];

x[j]=temp;

}

}

temp=x[pivot];

x[pivot]=x[j];

x[j]=temp;

quick\_sort(x,first,j-1);

quick\_sort(x,j+1,last);

}

}

int binary\_search(long int \*arr,int start,int end,int key,int d)

{

int mid;

if(start==end)

{

if(arr[start]==key)

return 1;

else

return -1;

}

else if(start<end)

{

mid=(start+end)/2;

if(arr[mid]==key)

return 1;

else if(arr[mid]<key)

binary\_search(arr,mid+1,end,key,d);

else

binary\_search(arr,start,mid-1,key,d);

}

}

int n\_ary\_search(long int \*arr,int start,int end,int key,int d)

{

long int mid[d];

mid[0]=start;

if(start>end)

{

return 0;

}

for(int i=1;i<=d-1;i++)

{

mid[i]=i\*(end-start)/d+start;

//printf("%d ",mid[i]);

}

for(int i=1;i<=d-1;i++)

{

if(arr[mid[i]]==key)

{

return 1;}

else if(key<arr[mid[i]])

{

n\_ary\_search(arr,mid[i-1],mid[i]-1,key,d);

return 0;

}

}

return n\_ary\_search(arr,mid[d-1]+1,end,key,d);

return 0;

}

int main()

{

FILE \*fp;

long int arr[6000],i,key,j,d,k,l,m,n,count=0;

clock\_t start,end;

fp=fopen("ne1.csv","w");

for(n=50;n<=500;n+=50)

{

fprintf(fp,"%ld ,",n);

//printf("Initialising %d elements...\t",n);

for(i=0;i<n;i++)

{arr[i]=rand()%10000;}

//printf("Initialisation ...[DONE]\nSORTING...\t");

quick\_sort(arr,0,n-1);

//printf("[DONE]\n");

//for(i=0;i<n;i++){

// printf("%d ",arr[i]);}

//printf("\n");

for(d=11;d>=2;d--)

{

start=clock();

for(k=0;k<500000;k++)

{

key=rand()%10000;

n\_ary\_search(arr,0,n-1,key,d);

count++;

//binary\_search(arr,0,n-1,key,d);

//printf("%d--%d. key=%d searched -- %d\n",n,d,key,k);

}

end=clock();

double JD=(double)end-(double)start;

printf("%f,",JD);

fprintf(fp,"%f,",JD);

//getch();

}

fprintf(fp,"\n");

}

fclose(fp);

//printf("%ld---end\n",count+1);

}