#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#define MAX 1000

int main()

{

int i,b;

FILE \*fp;

fp=fopen("data.txt","w");

for (b=0;b<MAX;b++)

{

i=rand()%32000;

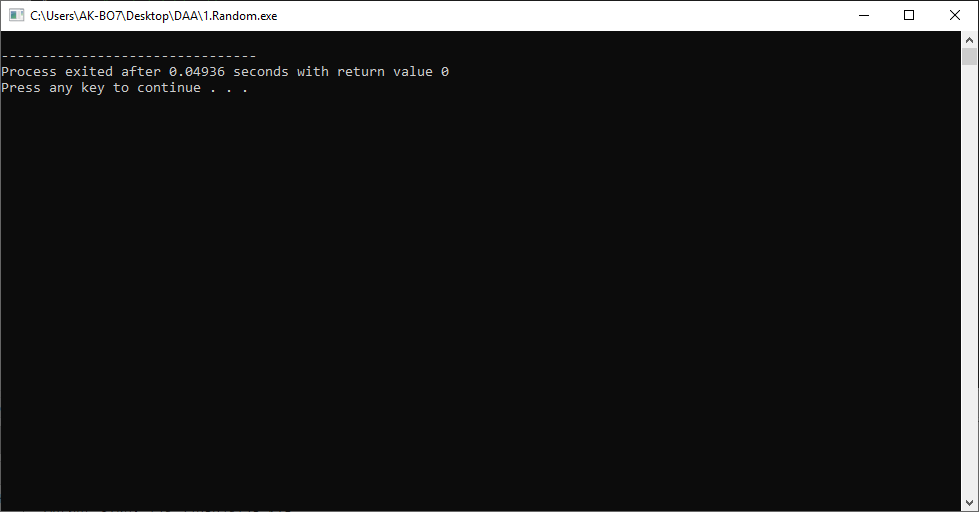
fprintf(fp, "%d\t",i);

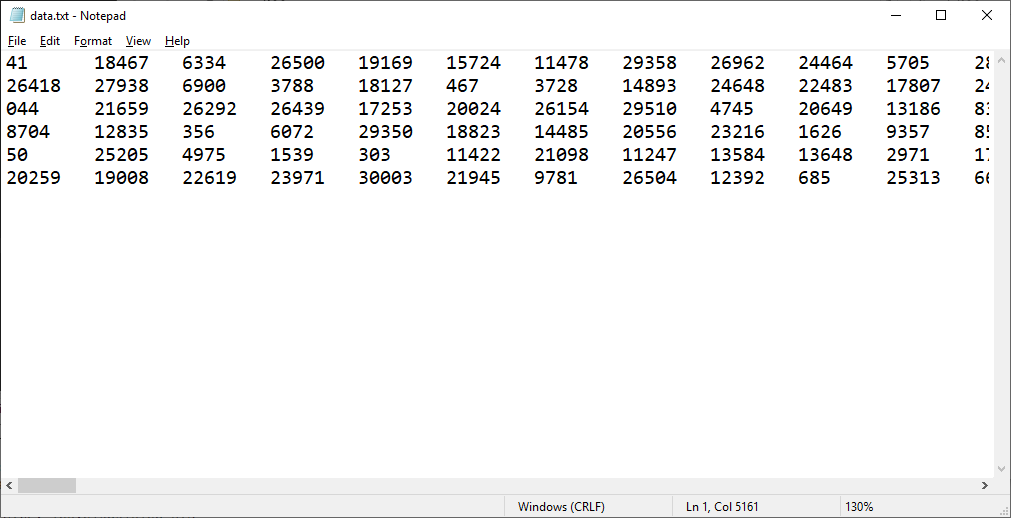
}

fclose(fp);

return 0;

}





2. heap sort

#include<conio.h>

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#define MAX 1000

double count=0,val,diff=0;

void randoms()

{

int i,b;

//double b;

FILE \*fp;

fp=fopen("data.txt","w");

for (b=0;b<MAX;b++)

{

i=rand()%32000;

fprintf(fp, "%d\n",i);

}

fclose(fp);

}

void getdata(int A[])

{

int i,b;

// double b;

FILE \*fp;

fp=fopen("data.txt","r");

for (b=0;b<MAX;b++)

{

fscanf(fp,"%d",&i);

A[b]=i;

}

fclose(fp);

}

void heapify(int arr[], int n, int i)

{ int temp=0;

int largest = i;

int l = 2\*i + 1;

int r = 2\*i + 2;

count++;

if (l < n && arr[l] > arr[largest])

largest = l; count++;

if (r < n && arr[r] > arr[largest])

largest = r; count++;

if (largest != i)

{

temp=arr[i];

arr[i]=arr[largest];

arr[largest]=temp;

heapify(arr, n, largest);

count++;

}

}

void heapSort(int b[], int n)

{

int temp=0,i;

for (i=((n/2)-1);i>=0;i--)

{heapify(b, n, i);

}

for (i=n-1; i>=0; i--)

{ count++;

temp=b[0];

b[0]=b[i];

b[i]=temp;

heapify(b, i, 0);

}

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

{

printf("%d\t",b[i]);}

//printf("\nTime taken= %lf",diff/(double)CLOCKS\_PER\_SEC);

printf("\n Comparisons= %lf",count);

}

int main()

{

int A[MAX];

int b;

randoms();

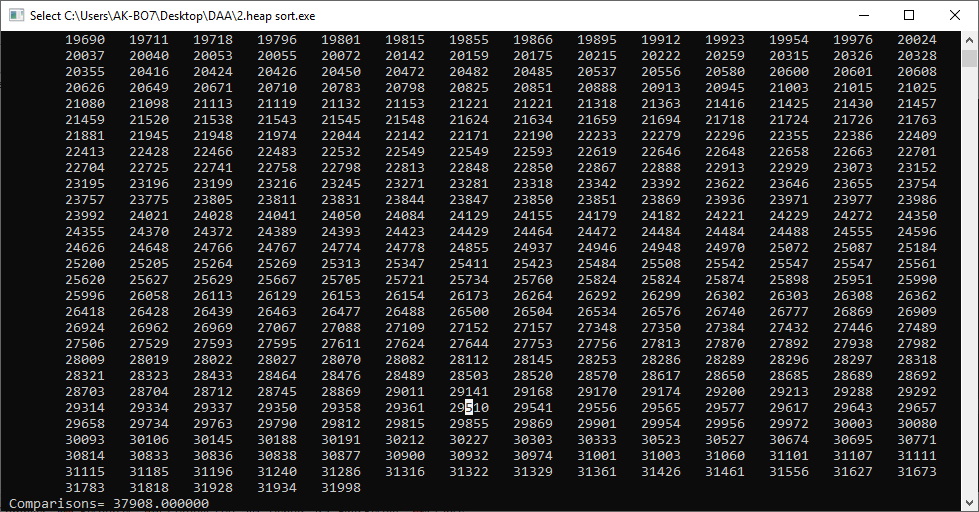
getdata(A);

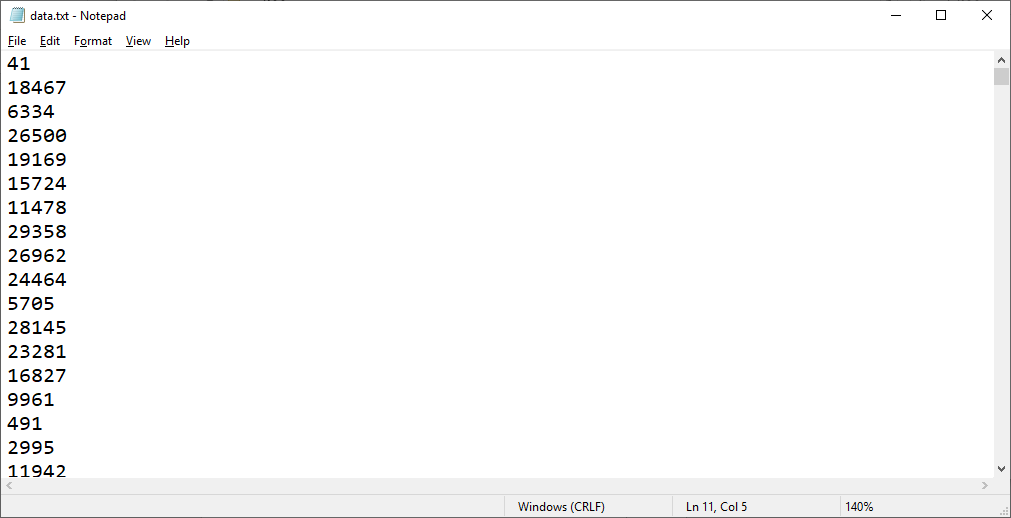
heapSort(A, MAX);

getch();

return 0;

}





Insertion sort

#include<conio.h>

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#define MAX 1000

void randoms()

{

int i,b;

//double b;

FILE \*fp;

fp=fopen("data.txt","w");

for (b=0;b<MAX;b++)

{

i=rand()%32000;

fprintf(fp, "%d\n",i);

}

fclose(fp);

}

void getdata(int A[])

{

int i,b;

// double b;

FILE \*fp;

fp=fopen("data.txt","r");

for (b=0;b<MAX;b++)

{

fscanf(fp,"%d",&i);

A[b]=i;

}

fclose(fp);

}

void inser\_sort(int A[])

{

int key,i,j,b;

double count=0,val,diff=0;

val=clock();

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

{

key=A[i];

j=i-1;

count++;

while(A[j]>=key&&j>=0)

{

A[j+1]=A[j];

j--;

count++;

}

A[j+1]=key;

}

diff=clock()-val;

for(b=0;b<MAX;b++)

printf("%d\t",A[b]);

printf("\nTime taken= %lf",diff/(double)CLOCKS\_PER\_SEC);

printf("\n Comparisons= %lf",count);

}

int main()

{

int A[MAX];

int b;

randoms();

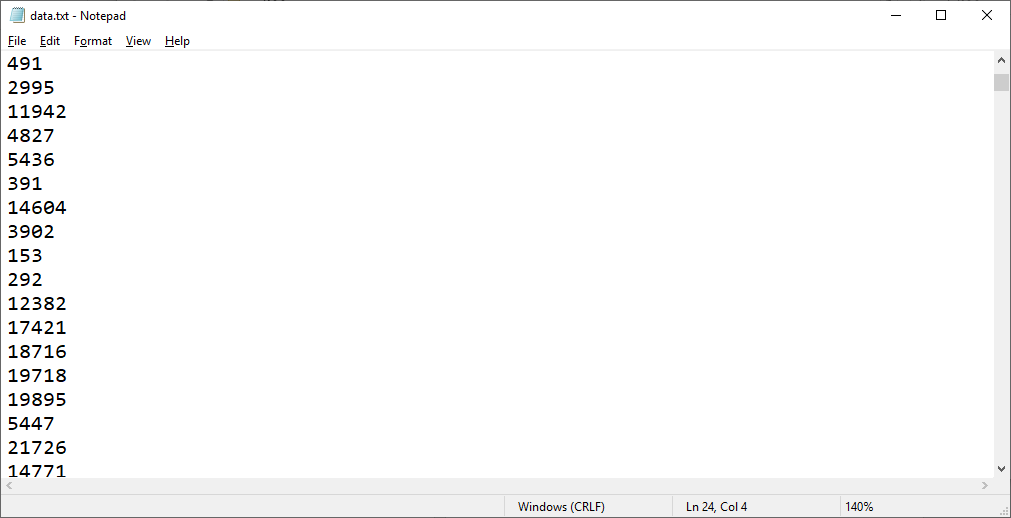
getdata(A);

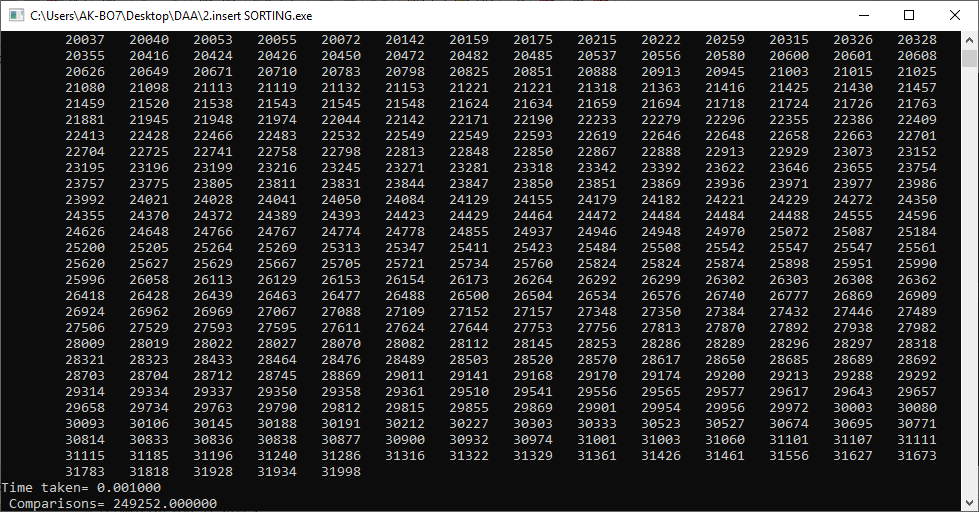
inser\_sort(A);

getch();

return 0;

}





Merge Sort

#include<conio.h>

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#define MAX 1000

double count=0,val,diff=0;

void randoms()

{

int i,b;

FILE \*fp;

fp=fopen("data.txt","w");

for (b=0;b<MAX;b++)

{

i=rand()%32000;

fprintf(fp, "%d\n",i);

}

fclose(fp);

}

void getdata(int A[])

{

int i,b;

// double b;

FILE \*fp;

fp=fopen("data.txt","r");

for (b=0;b<MAX;b++)

{

fscanf(fp,"%d",&i);

A[b]=i;

}

fclose(fp);

}

void merge(int Arr[], int start, int mid, int end)

{

int temp[end - start + 1];

int i = start, j = mid+1, k = 0;

while(i <= mid && j <= end) {

if(Arr[i] <= Arr[j]) {

count++;

temp[k] = Arr[i];

k += 1; i += 1;

}

else {

count++;

temp[k] = Arr[j];

k += 1; j += 1;

}

}

while(i <= mid) {count++;

temp[k] = Arr[i];

k += 1; i += 1;

}

while(j <= end) {count++;

temp[k] = Arr[j];

k += 1; j += 1;

}

for(i = start; i <= end; i += 1) {count++;

Arr[i] = temp[i - start];

}

}

void mergeSort(int Arr[], int start, int end) {

val=clock();

if(start < end) {

count++;

int mid = (start + end) / 2;

mergeSort(Arr, start, mid);

mergeSort(Arr, mid+1, end);

merge(Arr, start, mid, end);

}

}

void Sho(int A[])

{

diff=clock()-val;

for(int b=0;b<MAX;b++)

printf("%d\t",A[b]);

//printf("\nTime taken= %lf",diff/(double)CLOCKS\_PER\_SEC);

printf("\n Comparisons= %lf",count);

}

int main()

{

int A[MAX];

int b;

randoms();

getdata(A);

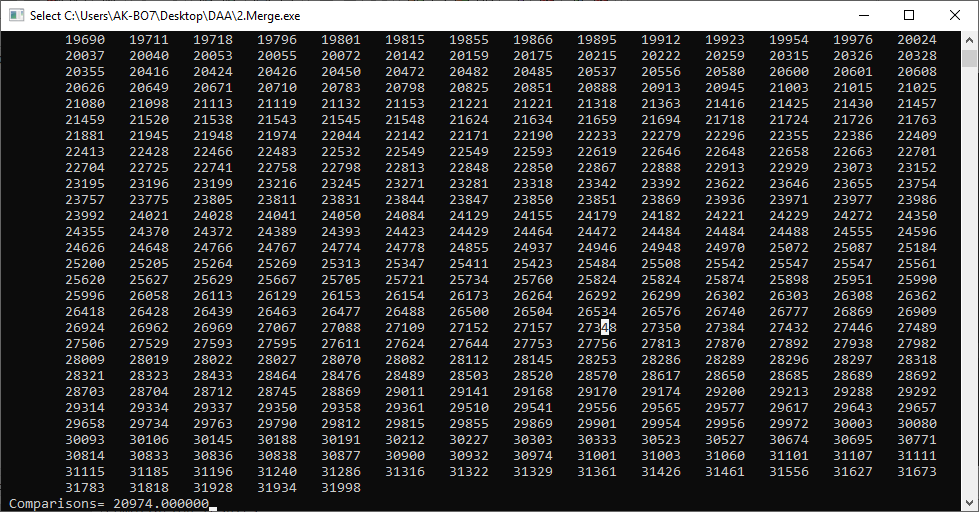
mergeSort(A,0,MAX);

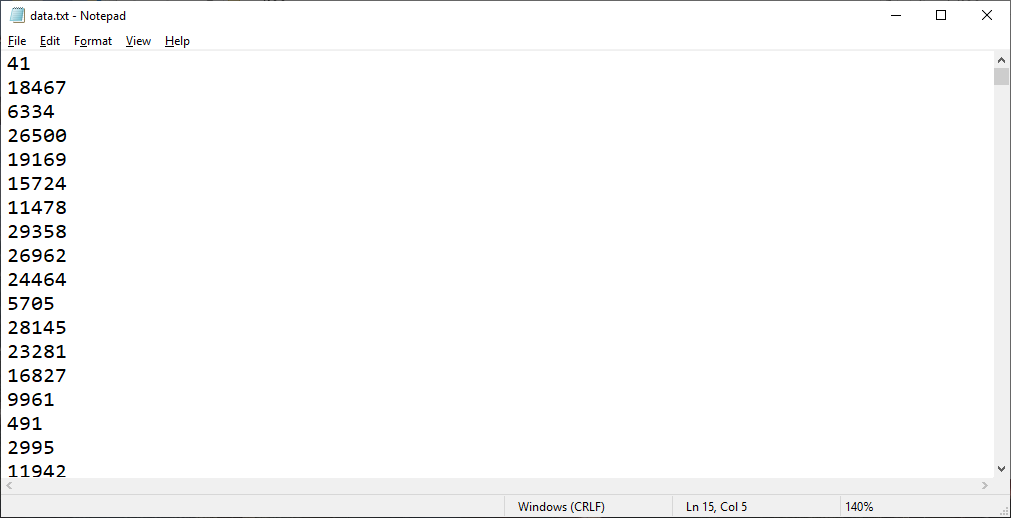
Sho(A);

getch();

return 0;

}





QUICK SORT

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<time.h>

#define MAX 1000

double count=0,val,diff=0;

void randoms()

{

int i,b;

//double b;

FILE \*fp;

fp=fopen("data.txt","w");

for (b=0;b<MAX;b++)

{

i=rand()%32000;

fprintf(fp, "%d\n",i);

}

fclose(fp);

}

void getdata(int A[])

{

int i,b;

// double b;

FILE \*fp;

fp=fopen("data.txt","r");

for (b=0;b<MAX;b++)

{

fscanf(fp,"%d",&i);

A[b]=i;

}

fclose(fp);

}

void swap(int \*p, int \*q)

{

int \*temp;

temp=p;

p=q;

q=temp;

}

int partition (int arr[], int low, int high)

{

int pivot = arr[high];

int i = (low - 1);

int j;

for ( j=low;j<=high-1;j++)

{ count++;

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)

{ count++;

int pi = partition(arr, low, high);

quickSort(arr, low, pi - 1);

quickSort(arr, pi + 1, high);

}

//diff=clock()-val;

}

int main()

{

int A[MAX];

int b;

randoms();

getdata(A);

quickSort(A,1,MAX);

for( b=0;b<MAX;b++)

{ printf("%d\t",A[b]);}

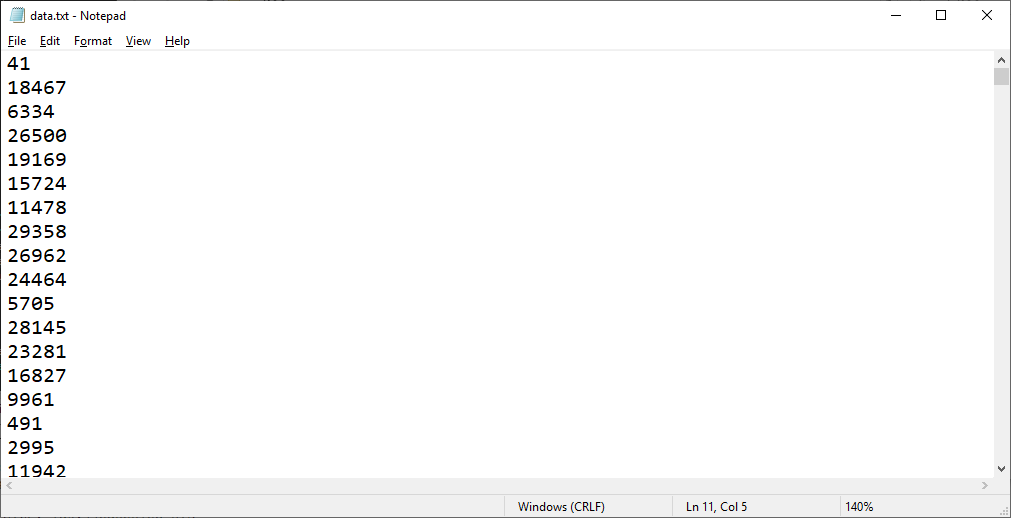
//printf("\nTime taken= %lf",diff/(double)CLOCKS\_PER\_SEC);

printf("\n Comparisons= %lf",count);

getch();

return 0;

}



3) comparison

#include<stdio.h>

#include<conio.h>

int min=32766 ,max=0,n,A[50],count=0;

int partion(int low,int high)

{ int i;

for( i=low;i<=high;i++)

{count++;

if(min>=A[i])

{

min=A[i];

}

if(max<=A[i])

{

max=A[i];

}

}

}

int main()

{ int i;

printf("\n Enter Size of Elements : ");

scanf("%d",&n);

printf("\n Enter Elements : ");

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

{scanf("%d",&A[i]);

}

if(n<2)

{count++;

min=A[0];

max=A[0];

}

else

{

partion(0,(n-1)/2);

partion((n-1)/2,n-1);

}

printf("\n Min=%d \t Max=%d \n",min,max);

printf("\n Comparion=%d ",count);

}

