Metoda Bulelor

**Metoda de sortare a bulelor (Bubble Sort) este unul dintre cele mai simple si intuitive algoritmi de sortare. Algoritmul functioneaza prin repetarea trecerilor prin vector, comparand perechi consecutive de elemente si interschimbandu-le daca acestea sunt in ordine gresita. In fiecare trecere, cel mai mare (sau cel mai mic, in cazul sortarii ascendente sau descendente) element este plasat in pozitia corespunzatoare. Aceasta procedura continua pana cand nu mai sunt necesare interschimbari, indicand ca vectorul este sortat.**

Se parcurge vectorul de la început până la sfârșit.

Pentru fiecare pereche de elemente consecutive, acestea sunt comparate.

Dacă elementul curent este mai mare sau mai mic decât elementul următor, pozitiile sunt schimbate.

Se continua acest proces pentru fiecare pereche de elemente de-a lungul întregii liste.

Se repeta pașii 1-4 până când nu mai sunt schimbări de poziții în timpul unei treceri prin vector.

Complexitatea timp a algoritmului de sortare a bulelor este, de asemenea, de O(n^2), unde "n" este numărul de elemente din lista de sortat. Acest lucru înseamnă că timpul necesar pentru sortarea unei liste crește în mod pătratic odată cu creșterea dimensiunii listei.

Probleme:

1. (509)-BULE

#include<iostream>

using namespace std;

int main(){

int v[101],n,aux;

bool sortat;

cin>>n;

for(int i=1;i<=n;i++){

cout<<"v["<<i<<"]="<<endl;

cin>>v[i];

}

do{

sortat=true;

for(int i=1;i<=n-1;i++){

if(v[i]>v[i+1]){

aux=v[i];

v[i]=v[i+1];

v[i+1]=aux;

sortat=false;

}

}

}while(!sortat);

for(int i=1;i<=n;i++)

cout<<v[i]<<" ";

return 0;

}

#include <iostream> --------4146

using namespace std;

int main()

{

int n,v[100],v2[100],poz,aux;

bool sortat;

cin>>n;

poz=1;

for(int i=1;i<=n;i++)

cin>>v[i];

for(int i=1;i<=n;i++)

{

if(v[i]%10==0){

v2[poz]=v[i];

poz++;

}

}

do

{

sortat = true;

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

if(v2[i] < v2[i+1])

{

int aux = v2[i];

v2[i] = v2[i+1];

v2[i+1] = aux;

sortat = false;

}

}

while(!sortat);

for(int i=1;i<poz;i++)

cout<<v2[i]<<" ";

return 0;}

#include <iostream> -----1608

using namespace std;

int main() {

int n,aux,aux2;

bool sortat;

cin>>n;

struct numar

{

int nr;

int divizori=0;

} nr[100];

for(int i=1;i<=n;i++)

{

cin>>nr[i].nr;

}

for(int i=1;i<=n;i++){

for(int j=1;j<=nr[i].nr;j++){

if(nr[i].nr%j==0)

nr[i].divizori++;

}

}

do{

sortat=true;

for(int i=1;i<=n;i++){

if(nr[i].divizori<nr[i+1].divizori){

aux=nr[i].divizori;

aux2=nr[i].nr;

nr[i].divizori=nr[i+1].divizori;

nr[i].nr=nr[i+1].nr;

nr[i+1].divizori=aux;

nr[i+1].nr=aux2;

sortat=false;

}

}

}while(!sortat);

for(int i=1;i<=n;i++)

cout<<"numarul "<<nr[i].nr<<" are "<<nr[i].divizori<<" divizori"<<endl;

return 0;

}

#include <iostream> ---------515

using namespace std;

int main() {

int n,s[100],x[100],aux;

bool sortat;

cin>>n;

for(int i=1;i<=n;i++){

cin>>x[i];

s[i]=0;

}

for(int i=1;i<=n;i++){

for(int j=1;j<=x[i];j++){

if(x[i]%j==0)

s[i]+=j;}

}

do{

sortat=true;

for(int i=1;i<n;i++){

if(s[i]>s[i+1]){

aux=s[i];

s[i]=s[i+1];

s[i+1]=aux;

aux=x[i];

x[i]=x[i+1];

x[i+1]=aux;

sortat = false;}

else

if(s[i]==s[i+1])

if(x[i]>x[i+1])

{

aux=x[i];

x[i]=x[i+1];

x[i+1]=aux;}

}

}while(!sortat);

for(int i=1;i<=n;i++)

cout<<x[i]<<"="<<s[i]<<endl;

return 0;

}

#include <iostream> ---------------BULE 186

using namespace std;

int main() {int n,nr[100],cifre[100],cif\_min;

bool sortat;

cin>>n;

for(int i=1;i<=n;i++)

cin>>nr[i];

for(int i=1;i<=n;i++){

cif\_min=nr[i]%10;

nr[i]/=10;

while(nr[i]!=0){

if(nr[i]%10<cif\_min)

cif\_min=nr[i]%10;

nr[i]/=10;}

cifre[i]=cif\_min;

cif\_min=9999;}

do{

sortat = true;

for(int i=1;i<n;i++)

if(cifre[i]>cifre[i+1]&&cifre[i]!=0){

int aux = cifre[i];

cifre[i] = cifre[i+1];

cifre[i+1] = aux;

sortat = false;}}

while(!sortat);

int i=1;

while (cifre[i]==0){

i++;}

if (cifre[1]==0){

cifre[1]=cifre[i];

cifre[i]=0;}

for(int i=1;i<=n;i++)

cout<<cifre[i];

return 0;

}

#include <iostream>

using namespace std;

int main() {

int n,nr[100],aux;

bool sortat;

cin>>n;

for(int i=1;i<=n;i++)

cin>>nr[i];

do{

sortat=true;

for(int i=1;i<n/2;i++){

if(nr[i]>nr[i+1]){

aux=nr[i];

nr[i]=nr[i+1];

nr[i+1]=aux;

sortat=false;

}

}

}while(!sortat);

for(int i=1;i<=n/2;i++){

cout<<nr[i];

}

do{

sortat=true;

for(int i=(n/2)+1;i<=n;i++){

if(nr[i]<nr[i+1]){

aux=nr[i];

nr[i]=nr[i+1];

nr[i+1]=aux;

sortat=false;

}

}

}while(!sortat);

for(int i=(n/2)+1;i<=n;i++){

cout<<nr[i];

}

return 0;

}

Selectia Directa

Metoda de sortare prin selecție directă este un algoritm simplu de sortare în care lista este împărțită în două părți: o parte sortată și o parte nesortată. Algoritmul selectează cel mai mic (sau cel mai mare, în funcție de dorință) element din partea nesortată și îl plasează în partea sortată. Acest proces este repetat până când toate elementele sunt plasate în ordine în partea sortată.

1.Inițial, lista este considerată ca fiind împărțită în două părți: o parte sortată (la început golă) și o parte nesortată (întreaga listă).

2.La fiecare pas, algoritmul găsește cel mai mic (sau cel mai mare) element din partea nesortată și îl plasează la sfârșitul părții sortate.

3.Procesul continuă până când toate elementele sunt sortate.

Complexitatea timp a algoritmului de sortare prin selecție directă este de O(n^2), unde "n" este numărul de elemente din lista de sortat. Aceasta înseamnă că timpul necesar pentru sortarea unei liste crește în mod pătratic odată cu creșterea dimensiunii listei.

Probleme:

#include<iostream> ------- SELECTIE DIRECTA

using namespace std;

int main(){

int v[101],n,aux;

cin>>n;

for(int i=1;i<=n;i++){

cout<<"v["<<i<<"]="<<endl;

cin>>v[i];

}

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

for(int j=i+1;j<=n;j++)

if(v[i]>v[j]){

aux=v[i];

v[i]=v[j];

v[j]=aux;

}

for(int i=1;i<=n;i++)

cout<<v[i]<<" ";

return 0;

}

#include <iostream> -----------------------510

using namespace std;

int main(){

int n,v[100],v2[100],poz,prim,aux;

bool sortat;

cin>>n;

poz=1;

for(int i=1;i<=n;i++)

cin>>v[i];

for(int i=1;i<=n;i++){

prim=1;

for(int j=2;j<v[i];j++){

if(v[i]%j==0 ){

prim=0;

break;}

}

if(prim==1 && v[i]!=1){

v2[poz]=v[i];

poz++;}

}

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

for(int j = i + 1 ; j < poz ; j ++)

if(v2[i] > v2[j]){

int aux = v2[i];

v2[i] = v2[j];

v2[j] = aux;

}

for(int i=1;i<poz;i++)

cout<<v2[i]<<" ";

return 0;}

#include <iostream> --------------------------------180

using namespace std;

int main() {

int n, nr[100], max = 1, min = 1;

cin >> n;

for (int i = 1; i <= n; i++) cin >> nr[i];

for (int i = 1; i <= n; i++) {

if (nr[i] > nr[max])

max = i;

else

if (nr[i] < nr[min])

min = i;

}

if (min > max) {

int aux = min;

min = max;

max = aux;

}

cout << "mmin=" << min << endl;

cout << "max=" << max << endl;

for (int i = min; i < max; i++)

for (int j = i + 1; j <= max; j++)

if (nr[i] > nr[j]) {

int aux = nr[i];

nr[i] = nr[j];

nr[j] = aux;

}

for (int i = 1; i <= n; i++) cout << nr[i] << " ";

return 0;

}

#include <iostream> -------------3403

using namespace std;

int main() {

int n,c,prod[100],aux,cod,disp=0,ndisp=0,cat,lista[100],coduri[100],poz=1,frecventa[100];

cin>>c;

cin>>n;

for(int i=1;i<=n;i++){

cout<<"produsul:"<<i<<" are codul: ";

cin>>prod[i];

cout<<endl;}

for(int j=1;j<=n;j++){

aux=prod[j];

while(aux!=0){

cod=cod\*10+aux%10;

aux/=10;}

if(((cod%100)/10)%2==0)

disp++;

else

ndisp++;

coduri[j]=cod;

cod=0;

frecventa[j]=0;}

if(c!=1){for(int j=1;j<=n;j++){

cat=coduri[j]%10;

if(frecventa[cat]==0){

cout<<cat<<":";

for(int i=1;i<=n;i++){

if(coduri[i]%10==cat){

lista[poz]=prod[i];

poz++;}}

for(int b= 1 ; b < poz-1 ; b ++)

for(int v = b + 1 ; v <poz ; v ++)

if(lista[b] > lista[v]){

int aux = lista[b];

lista[b] = lista[v];

lista[v] = aux;}

for(int k=1;k<poz;k++){cout<<lista[k]<<" ";lista[k]=0;}cout<<endl;poz=1;frecventa[cat]=1;}}}

if(c==1){cout<<disp<<endl;cout<<ndisp;}

return 0;}

#include <iostream> --------------185

using namespace std;

int main() {

int n,nr[100],cifre[100],aux;

cin>>n;

for(int i=1;i<=n;i++){

cin>>nr[i];

while(nr[i]>=10)

nr[i]/=10;

}

for(int i = 1 ; i < n ; i ++)

for(int j = i + 1 ; j <= n ; j ++)

if(nr[i] > nr[j])

{

aux = nr[i];

nr[i] = nr[j];

nr[j] = aux;

}

for(int i=1;i<=n;i++)

cout<<nr[i];

return 0;

}

#include <iostream> --------------------512

using namespace std;

int main() {

int n,nr[100],max=-1,maxpoz;

cin>>n;

for(int i=1;i<=n;i++){

cin>>nr[i];

if(nr[i]>max){

max=nr[i];

maxpoz=i;

}

}

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

for(int j = i + 1 ; j < maxpoz ; j ++)

if(nr[i] > nr[j])

{

int aux = nr[i];

nr[i] = nr[j];

nr[j] = aux;

}

for(int i=maxpoz+1;i<n;i++)

for(int j = i + 1 ; j <=n ; j ++)

if(nr[i] < nr[j]){

int aux = nr[i];

nr[i] = nr[j];

nr[j] = aux;}

for(int i=1;i<=n;i++)

cout<<nr[i]<<" ";

return 0;

}

Insertie

Sortarea prin inserție este un algoritm de sortare în care elementele sunt sortate una câte una, pe măsură ce sunt parcurse. Algoritmul începe prin a considera primul element ca fiind o secvență sortată (deoarece o secvență de un singur element este sortată în mod implicit) și apoi parcurge fiecare element din vector. Pentru fiecare element curent, acesta este inserat în secvența sortată așa încât secvența să rămână sortată. Procesul de inserție implică compararea elementului curent cu elementele din secvența sortată și mutarea elementelor mai mari către dreapta pentru a face loc elementului curent în poziția potrivită.

Pasii de executie:

1.Se consideră primul element ca fiind o secvență sortată.

2.Pentru fiecare element curent din vector:

- Se compară elementul curent cu elementele din secvența sortată.

- Se mută elementele mai mari către dreapta

pentru a face loc elementului curent.

- Se inserează elementul curent în poziția potrivită

astfel încât secvența să rămână sortată.

3.Procesul continuă până când toate elementele

din vector sunt sortate.

Sortarea prin inserție are o complexitate medie de timp de O(n^2), unde n este numărul de elemente din vector, deoarece trebuie să parcurgem întreaga secvență sortată pentru a găsi locul de inserție pentru fiecare element. Este un algoritm eficient pentru seturi de date mici sau aproape sortate, dar poate deveni ineficient pentru seturi de date mari.

Probleme:

#include<iostream> --------prin insertie

using namespace std;

int main(){

int v[101],n,aux;

cin>>n;

for(int i=1;i<=n;i++){

cout<<"v["<<i<<"]="<<endl;

cin>>v[i];

}

for(int i = 1 ; i <=n ; i ++)

{

int p = i;

while(p > 0 && v[p] < v[p-1])

{

int aux = v[p];

v[p] = v[p-1];

v[p-1] = aux;

p --;

}

}

for(int i=1;i<=n;i++)

cout<<v[i]<<" ";

return 0;}

#include <iostream> ---------------------618

using namespace std;

int main(){

struct elev{

int inaltime;

int pozitie;

} v[100];

int n,aux,aux2;

cin>>n;

for(int i=1;i<=n;i++){

cin>>v[i].inaltime;

v[i].pozitie=i;}

for(int i = 1 ; i <n ; i ++)

for(int j = i + 1 ; j <= n ; j ++)

if(v[i].inaltime > v[j].inaltime){

int aux = v[i].pozitie;

aux2 = v[i].inaltime;

v[i].pozitie = v[j].pozitie;

v[i].inaltime = v[j].inaltime;

v[j].pozitie = aux;

v[j].inaltime = aux2;}

for(int i=1;i<=n;i++)

cout<<v[i].pozitie<<"="<<v[i].inaltime<<" ";

return 0;}

#include <iostream> ------------------------605

using namespace std;

int main() {

int n,nr[100],poz[100],k;

cin>>n;

for(int i=1;i<=n;i++){

cin>>nr[i];

poz[i]=nr[i];

}

for(int i = 2 ; i <=n ; i ++)

{

int p = i;

while(p > 1 && nr[p] > nr[p-1]) // Modificare aici: de la a[p] < a[p-1] la a[p] > a[p-1]

{

int aux = nr[p];

nr[p] = nr[p-1];

nr[p-1] = aux;

p --;}

}

for(int i=1;i<=n;i++)

if(nr[i]==poz[i])

k++;

cout<<k;

return 0;}

#include <iostream> ----------------------2905

using namespace std;

int main() {

int n,c,p,nr[100],nrdiv[100],k,sort[100],poz=1;

cin>>c;

cin>>n;

cin>>p;

for(int i=1;i<=n;i++){

cin>>nr[i];

for(int j=1;j<=nr[i];j++)

if(nr[i]%j==0)

k++;

nrdiv[i]=k;

k=0;

}

if(c==2){

for(int i=1;i<=n;i++){

if(nrdiv[i]==nrdiv[p]){

sort[poz]=nr[i];

poz++;

k++;}

}

for(int i = 2 ; i <=k ; i ++){

int p = i;

while(p > 1 && sort[p] > sort[p-1]) // Modificare aici: de la a[p] < a[p-1] la a[p] > a[p-1]{

int aux = sort[p];

sort[p] = sort[p-1];

sort[p-1] = aux;

p --;}

}

for(int i=1;i<=k;i++)

cout<<sort[i]<<" ";}

if(c==1)

cout<<nrdiv[p];

return 0;}

#include<iostream> insertie 511

using namespace std;

int main(){

int n, nr[100], k;

cin >> n >> k;

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

cin >> nr[i];

for(int i = 1; i <= k; i++) {

int p = i;

while(p > 0 && nr[p] < nr[p - 1]) {

int aux = nr[p];

nr[p] = nr[p - 1];

nr[p - 1] = aux;

p--;

}

}

for(int i = k + 1; i < n; i++) {

int p = i;

while(p > k && nr[p] > nr[p - 1]) {

int aux = nr[p];

nr[p] = nr[p - 1];

nr[p - 1] = aux;

p--;

}

}

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

cout << nr[i] << " ";

return 0;

}

#include<iostream> ---------------------////////////////////130 insertie

using namespace std;

int main(){

int n,nr[100],s,svector[100];

cin>>n;

for(int i=1;i<=n;i++){

cin>>nr[i];

int aux=nr[i];

while(aux!=0){

s+=aux%10;

aux/=10;

}

svector[i]=s\*10+i;

s=0;

}

for(int i = 2 ; i <=n ; i ++)

{

int p = i;

while(p > 1 && svector[p] < svector[p-1])

{

int aux = svector[p];

svector[p] = svector[p-1];

svector[p-1] = aux;

p --;

}

}

for(int i=1;i<=n;i++)

cout<<nr[svector[i]%10]<<" ";

}