**SORTAREA VECTORILOR –EXEMPLE DE ALGORITMI DE SORTARE**

**Ce este sortarea?**

In informatica, „a sorta” inseamna a rearanja elementele dintr-o structura de date in ordine crescatoare (sau descrescatoare). De exemplu, daca avem vectorul:

| **Indice**  **Element(i)** | **0** | **1** | **2** | **3** | **4** | **5** |
| --- | --- | --- | --- | --- | --- | --- |
| **V[i](valoare el.)** | 27 | 14 | 77 | 85 | 34 | 16 |

| **Indice**  **element** | **0** | **1** | **2** | **3** | **4** | **5** |
| --- | --- | --- | --- | --- | --- | --- |
| **V[i](valoare el.)** | 14 | 16 | 27 | 34 | 77 | 85 |

Dupa ce vom rearanja elementele in ordine crescatoare vom obtine vectorul :

Sortarea este o operatie fundamentala in informatica.

* Putem sorta datele pentru prezentarea catre utilizator (de exemplu intr-o agenda telefonica, sau atunci cand navigam printre foldere, sau selectam o anumita melodie din telefon).
* Sortarea este o piatra fundamentala in algoritmii mai mari, pentru ca simplifica anumite cerinte (precum gasirea unei chei unice, sau detectarea elementelor duplicate).
* Putem sorta orice, numere, texte, orice care are un criteriu de sortare. Dar pentru usurinta vom folosi numere in toate exemplele noastre.
* Trebuie sa avem grije sa distingem pozitia elementului (**i)** de valoarea efectiva a elementului ( **V[i] )**

**Sunt cunoscuti o multitudine de algoritmi de sortare , dar vom exemplifica pentru inceput doar doi.**

1. **Sortarea prin selecţie directă**

**Considerăm un vector de elemente comparabile între ele şi dorim să le ordonăm crescător. Pentru aceasta comparăm primul element cu toate elementele care urmează după el. Dacă găsim un element mai mic decât primul atunci le interschimbăm pe cele două. Apoi continuăm cu al doilea element al şirului, pe care, de asemenea îl comparăm cu toate elementele care urmează după el şi în caz de inversiune interschimbăm cele două elemente. Apoi procedăm la fel cu al treilea element al şirului iar procesul continuă astfel pâna la penultimul element al şirului care va fi comparat cu ultimul element din şir.**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int i,j,n,aux,a[50];**

**cout<<"Introduceti numarul de elemente din sir : ";**

**cin>>n; cout<<"Introduceti numerele"<<endl;**

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

**{**

**cout<<"a["<<i<<"]=";**

**cin>>a[i];**

**}**

**//urmeaza algoritmul de sortare**

**for(i=0;i<n-1;i++)**

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

**if (a[j]<a[i])**

**{**

**aux=a[i];**

**a[i]=a[j];**

**a[j]=aux;**

**}**

**//urmeaza afisarea sirului sortat**

**cout<<"Sirul sortat este:"<<endl;**

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

**cout<<a[i]<<" ";**

**cout<<endl;**

**}**

**2.** **Sortarea prin metoda bulelor  
  
   Acest algoritm se mai numeşte şi "sortarea prin selecţie şi interschimbare", "sortarea prin propagare" sau "metoda lentă de sortare" datorită numărului mare de operaţii care trebuie efectuate. Succesul algoritmului este asigurat de trecerea succesivă prin tablou, până când acesta este sortat, cu specificaţia că, la fiecare trecere, elementele succesive i şi i+1 pentru care a[i]>a[i+1], vor fi interschimbate. Metoda poate fi imbunătăţită dacă, după fiecare trecere, se va reţine ultima poziţie din tablou în care a avut loc o interschimbare, iar trecerea următoare se va efectua doar pină la acea poziţie. În cazul în care la o trecere nu a avut loc nici o interschimbare algoritmul se va incheia.**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int n,i,aux,terminat,a[50];**

**cout<<"Introduceti dimensiunea vectorului : ";**

**cin>>n;**

**for(i=0;i<=n-1;i++) {**

**cout<<"a["<<i<<"]=";**

**cin>>a[i];}**

**do**

**{**

**terminat=1;//se presupune ca vectorul este ordonat , dar se verifica**

**for(i=0;i<n-1;i++)**

**if(a[i]>a[i+1])**

**{**

**aux=a[i];**

**a[i]=a[i+1];**

**a[i+1]=aux;**

**terminat=0;}**

**}while (terminat==0);**

**cout<<"Vectorul ordonat este : ";**

**for(i=0;i<=n-1;i++)**

**cout<<a[i]<<" ";**

**}**

TEMA

De scris in caietele voastre de informatica intreg materialul de mai sus si de rezolvat singuri algoritmii echivalenti pentru ordonarea descrescatoare .