

Vectorul de frecvențe și aplicații

Vectorul de frecvențe reține numărul de apariții al fiecărei valori citite într-un vector. Folosirea vectorului de frecvențe permite scrierea unor algoritmi eficienți în cazul în care datele de intrare au valori dintr-un domeniu cunoscut care poate fi prelucrat rapid. Folosirea unui vector de frecvență sau marcarea este eficientă numai în cazul în care valorile care interesează sunt întregi și numărul valorilor distincte posibile este cel mult 1.000.000 pentru un timp maxim de 1 sec/test.

De exemplu acest vector poate fi folosit pentru sortarea rapidă a datelor, în timp liniar.

Tot cu ajutorul acestui vector pot fi implementate operațiile de bază cu mulțimi:

- Căutarea unui element
- Intersecția a două (sau mai multe) mulțimi
- Reuniunea a două (sau mai multe) mulțimi

Exemplu

Să considerăm din nou un șir de cifre zecimale. Să se determine cifra care apare de cele mai multe ori. Dacă sunt mai multe cifre care apar de număr maxim de ori, să se determine cea mai mică (sau cea mai mare, sau toate, etc.).

Putem folosi un **vector de frecvență**, adică un vector cu același număr de elemente ca numărul posibil de valori distincte din șirul dat (adică 10), cu semnificația: $v[c] =$ numărul de apariții ale cifrei c în șirul dat.

După construirea acestui vector, valoarea maximă din el va reprezenta numărul maxim de apariții ale unei cifre, iar indicii pentru care elementul corespunzător are valoare maximă reprezintă cifrele care apar de număr maxim de ori.

Exemplu:

$x[]$ 7 9 0 4 7 4 3 7

0	1	2	3	4	5	6	7	8	9
1	0	0	1	2	0	0	2	0	1

$v[]$

Probleme care se rezolva folosind vectori de frecventa

1. Se dau n numere naturale cu cel mult două cifre fiecare. Să se determine acele numere care apar o singură dată.

```
#include <iostream>
using namespace std;
int n,x,fr[100];
int main()
{
    cin>>n;
    for(int i=0;i<n;i++)
    {cin>>x;
    fr[x]++;
    }
    for(int i=0;i<100;i++)
        if(fr[i]==1)cout<<i<<" ";
    return 0;
}
```

2. Să se scrie un program care citește cel mult 1000000 de numere naturale din intervalul închis $[0,9]$ din fisierul intrare.txt și determină cel mai mare număr prim citit și numărul său de apariții.

```
#include <iostream>
#include <fstream>
using namespace std;
int fr[10],n;
int main()
{
    ifstream f("intrare.txt");
    while( f>>n) fr[n]++;
    if(fr[7]!=0)cout<<7<<" "<<fr[7];
    else
        if(fr[5] != 0)cout<<5<<" "<<fr[5];
        else
            if(fr[3] != 0 )cout<<3<<" "<<fr[3];
            else
                if(fr[2] != 0 )cout<<2<<" "<<fr[2];

    return 0;
}
```

3. Se citeste din fisierul text intrare.txt cel putin 3 si cel mult 1.000.000 de numere naturale din intervalul (0, 1.000.000.000). Se cere sa se afiseze pe ecran, separate printr-un spatiu, doua numere distincte, anume cel mai mare numar impar cu doua cifre si cel mai mic numar par cu doua cifre care NU fac parte din sir. Daca nu exista doua astfel de valori se va afisa pe ecran mesajul nu exista.

```
#include <iostream>
#include <fstream>
using namespace std;
int fr[100],x,ok;
int main()
{
    int n;
    ifstream f("intrare.txt");

    while (f>>x)fr[x] ++;
    int a=0, b=0 ;

    for(int  x=99 ; x > 10 ; x -=2)
        if ( fr[x]== 0 )
            {a = x ;
             ok ++ ;
             break ;
            }
    for( int x = 10 ; x < 99 ; x += 2)
        if ( fr[x] == 0 )
            { b = x ;
             ok ++ ;
             break ; }
    if ( ok == 2 )cout<< a << " " << b ;
    else cout<< " nu exista" ;

    return 0 ;
}
```

4. Să se scrie un program care citește cel mult 1000000 de numere naturale din intervalul închis $[0,9]$ din fisierul `intrare.txt` și afiseaza numerele care apar in fisier in ordine crescatoare

```
#include <iostream>
#include <fstream>
using namespace std;
int fr[100],x,i,j;
int main()
{
    int n;
    ifstream f("intrare.txt");

    while (f>>x)fr[x]++;
    for (i=0;i<=9;i++)
        for(j=1;j<=fr[i];j++) cout<<i<<' ';
    return 0 ;
}
```

5. Să se scrie un program care citește cel mult 1000000 de numere naturale din intervalul închis $[0,9]$ din fisierul `intrare.txt` și afiseaza numerele distincte care apar in fisier in ordine crescatoare

```
#include <iostream>
#include <fstream>
using namespace std;
int fr[100],x,i,j;
int main()
{
    int n;
    ifstream f("intrare.txt");

    while (f>>x)fr[x]++;
    for (i=0;i<=9;i++)
        if (fr[i]!=0) cout<<i<<' ';
    return 0 ;
}
```