

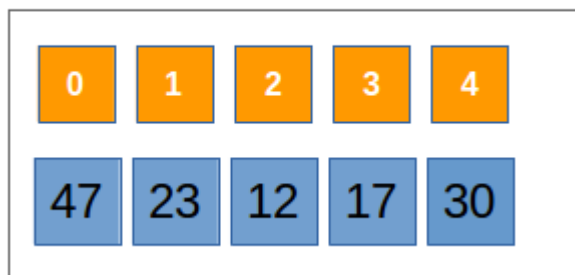
**Metoda bulelor** cunoscută și sub numele *BubbleSort*, metoda bulelor se bazează pe următoare idee:

- fie un vector `x[]` cu `n` elemente
- parcurgem vectorul și pentru oricare două elemente învecinate care nu sunt în ordinea dorită, le interschimbăm valorile
- după o singură parcurgere, vectorul nu se va sorta, dar putem repeta parcurgerea
- dacă la o parcurgere nu se face nicio interschimbare, vectorul este sortat

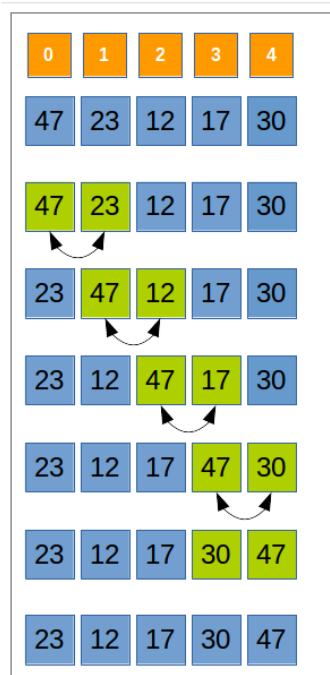
O reprezentare a algoritmului este:

- cat timp vectorul nu este sortat
  - presupunem că vectorul este sortat
  - parcurgem vectorul
    - dacă două elemente învecinate nu sunt în ordinea dorită
      - le interschimbăm
      - schimbăm presupunerea inițială

**Exemplu:** Să ordonăm următorul vector, în care `n=5`:

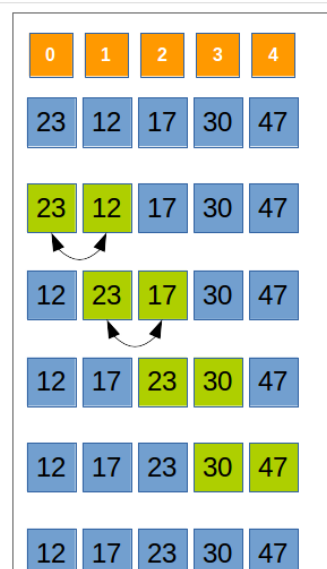


După prima parcurgere vectorul are următoarea structură:



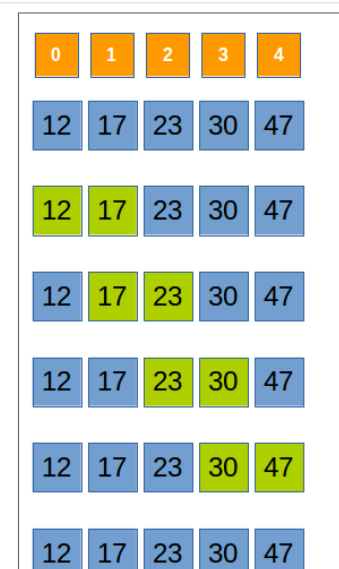
Deoarece am făcut interschimbări, nu știm dacă vectorul este ordonat.

După o nouă parcurgere, vectorul este:



Și acum am făcut interschimbări, nu știm dacă vectorul este ordonat (chiar dacă el este).

Îl parcurgem din nou:



Deoarece nu am făcut nicio interschimbare, decidem că vectorul este ordonat.

## O secvență C:

```
int n, v[100];
//citire v[] cu n elemente
bool sortat;
do
{
    sortat = true;
    for(int i = 0 ; i < n - 1 ; i ++)
        if(v[i] > v[i+1])
        {
            int aux = v[i];
            v[i] = v[i+1];
            v[i+1] = aux;
            sortat = false;
        }
}
while(!sortat);
```

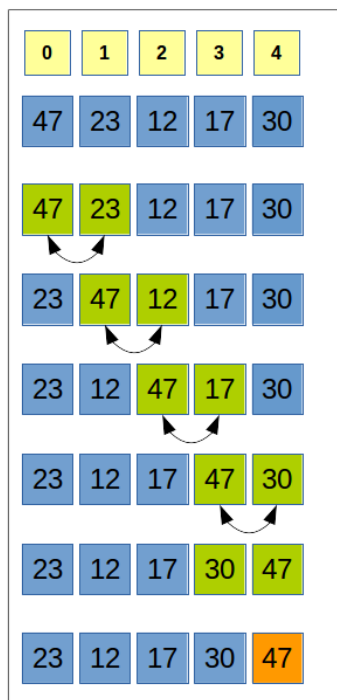
**Observație:** La fiecare parcurgere cel puțin un element ajunge pe poziția sa finală:

- la prima parcurgere cel mai mare element al vectorului ajunge pe poziția sa finală;
- la a doua parcurgere următorul cel mai mare element ajunge pe poziția finală;

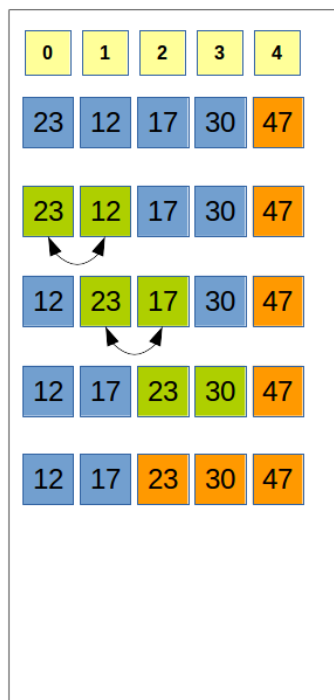
Observăm că nu mai are rost să parcurgem aceste elemente, fixate. Astfel, putem parcurge vectorul numai până la indicele unde s-a făcut ultima interschimbare la parcurgerea anterioară.

Exemplu:

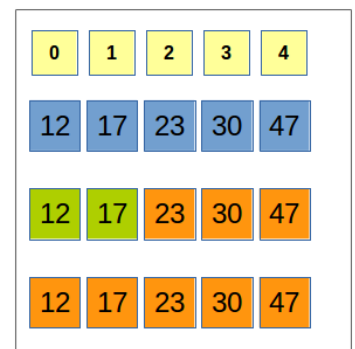
Parcursere 1



Parcursere 2



Parcursere 3



### O secvență C++:

```
int n, v[100];
//citire v[] cu n elemente
bool sortat;
int m = n;
do
{
    sortat = true;
    int p = m;
    for(int i = 0 ; i < p - 1 ; i++)
        if(v[i] > v[i+1])
        {
            int aux = v[i];
            v[i] = v[i+1];
            v[i+1] = aux;
            sortat = false;
            m = i + 1;
        }
}
while(!sortat);
```

