

## Interclasarea

Este algoritmul optim prin care, din elementele a doi vectori **sortați**  $a$  (cu  $n$  elemente) și  $b$  (cu  $m$  elemente) obținem un vector care este, de asemenea, sortat.

Algoritmul va menține un indice curent  $i$  în primul vector și un indice curent  $j$  în al doilea vector. Inițial  $i = j = 1$ .

La fiecare pas se compară elementul curent din primul vector cu elementul curent din al doilea, trecându-se în vectorul final cel mai mic dintre ele și incrementând indicele corespunzător.

La terminarea elementelor dintr-unul dintre vectori, toate elementele rămase în celălalt vector se vor trece în vectorul final.

Țineți scrierea sa în C++. Vectorul final va fi  $\leq$  cu  $k$  elemente.

```
i = j = 1; k = 0;
while (i <= n & j <= m) // deci mai 15 elemente în ambele
    if (a[i] < b[j])
        c[++k] = a[i++];
    else
        c[++k] = b[j++];
while (i <= n) // punem tot ce-a rămas în vectorul
    c[++k] = a[i++]; // a, dacă e cazul
while (j <= m) // analog pt. vectorul b
    c[++k] = b[j++];
```

Obs: Dacă există elemente egale, acestea se vor prelucra în totalitate.

---

### Căutarea binară

Este algoritmul care permite găsirea unui anumit element într-un vector **SORTAT** într-un număr foarte mic de pași.

Algoritmul se bazează pe restrângerea intervalului indicator elementelor între care facem căutarea a.î. numărul acestora se înjumătățește la fiecare pas.