Lectia #2 Algoritmi pe vectori (Sortare, cautare binara, sume partiale, interclasare)

Sortarea unui vector

Sortarea unui vector reprezinta rearanjarea elementelor astfel incat ele sa fie intr-o anumita ordine. Ex: sortare crescatoare, sortare descrescatoare.
(Diferiti algoritmi de sortare: https://www.geeksforgeeks.org/sorting-algorithms/). O sa urmarim Bubble sort si functia sort() din STL.

Cautare Binara

Căutarea unei valori într-un vector se poate face în două moduri:

- **secvențial** – presupune analizarea fiecărui element al vectorului într-o anumită ordine (de obicei de la stânga la dreapta). Când se găsește valoarea căutată parcurgerea vectorului se poate opri. În cel mai rău caz, pentru un vector cu `n` elemente parcurgerea face `n` pași, complexitatea timp a căutării secvențiale este `O(n)`
- **binar**. Căutarea binară se poate face într-un vector **numai dacă elementele acestuia sunt în ordine** (de obicei crescătoare) după un anumit criteriu (de obicei criteriul este chiar relația de ordine naturală între numere, cuvinte, etc). Căutarea binară presupune împărțirea vectorului în secvențe din ce în ce mai mici, înjumătățindu-le și continuând cu jumătatea în care se poate afla valoarea dorită (conform ordinii elementelor din vector). În cel mai rău caz, pentru un vector cu `n` elemente parcurgerea face `log2(n)` pași, complexitatea timp a căutării binare este `O(log2(n)).`

Q: De ce am folosi cautarea binara?

A: Este mult mai eficienta!!!!!

Exemplu: Avem de cautat, pe rand, m (m <= 10^7) valori intr-un vector de n (n <= 10 ^ 7) elemente. Daca am face o cautare secventiala pentru fiecare dintre cele m valori, am face m * n pasi in total, in cel mai rau caz. Deci **10^7 * 10^7** operatii. Daca, in schimb am folosi cautarea binara, am ajunge la doar m * log2(n) operatii, adica, in cel mai rau caz, 10^7 * log2(10 ^ 7)

(aprox. **10^7 * 24**). Astfel, obtinem un algoritm de aprox. 400.000 de ori mai eficient!

Probleme Cautare Binara

- https://www.pbinfo.ro/probleme/508/cautare-binara
- https://www.pbinfo.ro/probleme/2276/cb
- https://www.pbinfo.ro/probleme/536/fabrica1
- Pentru cine doreste probleme extra: https://usaco.guide/silver/binary-search

Sume partiale in vectori

In anumite probleme, vom avea de determinat rapid suma elementelor dintr-un anumit interval. Desigur aceasta se poate determina parcurgand elementele din intervalul respectiv, dar astfel obtinem o complexitate de O(n). Pentru a reduce numarul de operatii, putem folosi sume partiale.

Conceptul de sume partiale:

- Presupunem ca avem un vector a[] de n numere naturale.
- Vom crea un vector aditional s[], cu semnificatia s[i] = suma tuturor numerelor din a[], pana la pozitia i.
- Pe s[i] il putem obtine dupa formula urmatoare, s[i] = s[i - 1] + a[i], i > 0, s[0] = a[0].
- Dupa crearea vectorului s[], putem obtine suma elementelor de pe un anumit interval [i, j]

  dupa urmatoarea formula: suma(i, j) = s[j] - s[i - 1].

Probleme cu sume partiale

- https://www.usaco.org/index.php?page=viewproblem2&cpid=572
- https://www.usaco.org/index.php?page=viewproblem2&cpid=595
- O problema mai challenging: https://atcoder.jp/contests/abc125/tasks/abc125_c (hotaram daca o facem la ora sau o lasam optionala pentru acasa)

Interclasare

Considerăm două vectori cu elemente numere întregi **ordonate crescător**. Se dorește construirea unui alt vector, care să conțină valorile din cele două tablouri, în ordine.

Pentru rezolvarea acestei probleme putem folosi **interclasarea**.

Descrierea conceptului:

- Fie 2 vectori a[] de lungime n, respectiv b[] de lungime m
- Parcurgem in acelasi timp ambii vectori
- La fiecare pas:
  - alegem valoarea cea mai mica dintre cele doua elemente curente si o punem in cel de-al treilea vector
  - Avansam doar in tabloul din care am ales valoarea minima
- Odata ce se incheie parcurgerea unuia dintre tablouri (la pasul i = min(n, m)) adaugam toate elementele ramase in vectorul de lungime mai mare in vectorul destinatie.

Intrebari:

1. Daca am aplicat corect algoritmul, cate elemente vom avea in vectorul destinatie?
2. Cati pasi face algoritmul descris?

Probleme interclasare

- https://www.pbinfo.ro/probleme/241/interclasare
- https://www.pbinfo.ro/probleme/530/multimi1
- https://www.pbinfo.ro/probleme/2668/comun
- https://www.pbinfo.ro/probleme/3960/intersectie-siruri (optionala, este mult mai challenging).