

## Prezentare soluție concurs programare

Pentru rezolvarea acestei probleme am încercat să găesc cea mai practică și logică soluție.

În prima fază sunt citite și memorate datele problemei. Apoi, este parcursă lista comenzilor clienților și este găsită cea mai mare comandă (căci este mai bine să se găsească prețul optim pentru comenzile cu cea mai mare cantitate de unități). Pentru fiecare client este găsită combinația fabricant-depozit care să ofere costul total cel mai mic, indiferent dacă fabricantul poate onora întreaga comandă. Acest lucru produce un cost total mai mare în cazurile în care un fabricant nu poate onora întreaga comandă, iar liniile de transport folosite nu mai sunt utilizate pentru nicio altă comandă. În schimb, costul total produs este mai mic dacă oricare dintre aceste linii de transport este utilizată de o altă comandă.

Acest lucru se întâmplă deoarece se ține cont de liniile de transport deja folosite, mai multe transporturi de la un fabricant la un depozit putând fi considerate un singur transport mare, cu toate că destinațiile finale (clienții) diferă. Același lucru se întâmplă și în cazul drumurilor de la depozit la client. În practică, după ce se adună la cost un element din **Fij**, respectiv **Fjk**, acesta devine 0 în matrice.

Pentru fiecare transport calcule se ține cont de câte produse rămân în afara unui set de 100 pentru fiecare linie, căci din nou, dacă doi clienți primesc produse de la aceeași combinație fabricant-depozit, prima fază a transportului este comasată. Asemenea și pentru a doua fază a transportului.

Din punct de vedere al alegerilor tehnice, citind problema și având în vedere experiența mea cu limbajele de programare, am fost nehotărât între C/ C++ și Python. Prefer C/ C++ pentru probleme de algoritmică, dar consider listele din Python utile. Am găsit o soluție pentru această dilemă folosind container-ul **vector** din **Standard Template Library**, care funcționează similar cu listele din Python. Fiind pregătit pentru un set mare de date (valori pozitive) și neavând restricții de memorie, am folosit preponderent tipul de variabilă **unsigned long long**.

Consider că puteam îmbunătăți rezolvarea în diferite moduri. O îmbunătățire constă în verificarea transportului de cost minim în caz că acesta nu satisface comanda în totalitate (cazul în care diferența de preț este mică, dar diferența de cantitate este mare).

În concluzie, cu toate că observ modurile în care soluția dezvoltată poate fi îmbunătățită și schimbată, sunt mulțumit de aceasta.