

 $O(?)...$ 

# Metode de REDUCERE a COMPLEXITĂȚII

Mihai Stroe

**În cadrul acestui articol vom prezenta câteva tehnici cu ajutorul cărora se poate face trecerea, pentru anumite tipuri de probleme, de la complexitatea liniară a unor pași din rezolvare la complexitatea logaritmică, folosind principiul căutării binare.**

În ultimii ani au început să apară multe probleme care se pot rezolva folosind algoritmi al căror ordin de complexitate este  $O(N \cdot \log N)$  sau  $O(N^2 \cdot \log N)$ . Câteva rezolvări ale unor probleme clasice de acest gen (de exemplu, problema determinării subșirului crescător de lungime maximă) sunt cunoscute. Alte rezolvări se bazează pe structuri de date speciale (*heap*-uri, arbori AVL, arbori de intervale etc.)

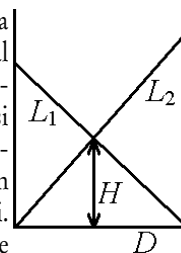
O categorie de probleme este considerată mai aparte deoarece, la un moment dat, în rezolvarea lor se introduce căutarea binară. Vom examina câteva astfel de probleme. Cititorul este invitat să încerce să rezolve problemele înainte de a citi sugestiile autorului.

Enunțurile problemelor pe care le vom examina sunt următoarele:

- Se consideră un șir de  $N$  numere naturale și un număr natural  $S$ . Se cere să se aleagă trei numere din acest șir, astfel încât suma lor să fie egală cu  $S$ .
- Se consideră un labirint sub forma unei matrice pătratică de dimensiune  $N$  ale cărei elemente pot lua valorile 0 sau 1, un punct de plecare și un punct de sosire. Dacă un element din matrice are valoarea 1, înseamnă că pe poziția corespunzătoare din labirint este zid, sau spațiu liber în caz contrar. Se cere să se afișeze dimensiunea celui mai mare pătrat care poate fi deplasat de la punctul de plecare la punctul de sosire. Un pătrat de latura  $K$  va ocupa în fiecare moment  $K^2$  poziții din matrice. Pătratul se poate deplasa pe orizontală și verticală și nu poate trece prin ziduri.
- Se consideră un graf neorientat. Fiecare muchie are asociată o lungime și un grad de periculozitate. Se cere să se afișeze un drum de la nodul 1 la nodul  $N$ , astfel încât gradul de periculozitate al celei mai periculoase muchii să fie cât mai mic posibil; dacă există mai multe drumuri, se va afișa drumul care are cea mai mică lungime.
- Se consideră un arbore cu  $N$  noduri ( $N \leq 100.000$ ). În cadrul său se pot alege  $K$  baze, situate în  $K$  noduri. Pentru fiecare nod se calculează distanța până la cea mai

apropiată bază. Determinați o modalitate de alegere a bazelor, astfel încât cea mai mare dintre distanțele nodurilor față de cea mai apropiată bază să fie cât mai mică posibil.

- Doi pereți verticali se află la o distanță  $D$  unul de celălalt. O scândură de lungime  $L_1$  pleacă de la baza primului perete și se sprijină pe al doilea. O scândură de lungime  $L_2$  pleacă de la baza celui de-al doilea perete și se sprijină pe primul. Cele două scânduri se ating la înălțimea  $H$ , într-un punct situat undeva între cei doi pereți. Dându-se  $L_1$ ,  $L_2$  și  $H$ , să se calculeze  $D$ ; este permisă o eroare de  $10^{-5}$ .



Prima problemă poate fi rezolvată prin metoda "forței brute" folosind trei cicluri repetitive **for**. Din nefericire, valoarea pentru  $N$  s-ar putea să fie prea mare pentru ca această rezolvare să se încadreze în timp.

Soluția propusă care folosește căutarea binară este următoarea: se sortează șirul, se folosesc două cicluri repetitive **for** pentru a alege două numere (fie acestea  $N_1$  și  $N_2$ ), apoi se caută binar valoarea  $S - N_1 - N_2$ . Această rezolvare are ordinul de complexitate  $O(N^2 \cdot 2 \log N)$  și mai poate fi optimizată (de exemplu, dacă valoarea  $N_1$  depășește valoarea  $S$  se iese din primul ciclu etc.). Varianta optimizată va avea șanse mari să treacă toate testele.

Invităm cititorul să găsească o rezolvare în  $O(N^2)$ , fără a folosi căutarea binară.

Cea de-a doua problemă este mai dificilă.

Dacă pătratul ar avea latura 1, folosirea algoritmului lui Lee ar rezolva problema. Acest algoritm va fi folosit oricum, dar dificultatea constă, la fiecare pas, în evitarea atingerii unui zid. Dacă se deplasează un pătrat de latură  $K$ , nu se dorește examinarea a  $K$  poziții vecine pentru, să zicem, o deplasare la dreapta. Aceasta se poate rezolva



folosind sume parțiale pe linii și pe coloane; o altă modalitate constă în precalcularea, pentru fiecare poziție, a laturii maxime a unui pătrat care poate fi deplasat din poziția respectivă pe fiecare din cele patru direcții.

Până în acest punct, algoritmul constă în încercarea deplasării unui pătrat de latură 1, 2, 3 etc., până când deplasarea nu mai reușește; se alege cea mai mare dimensiune pentru care deplasarea a reușit. O îmbunătățire ar putea consta în folosirea căutării binare pentru lungimea laturii. Astfel, inițial se va încerca deplasarea unui pătrat de dimensiune  $N/2$ ; dacă aceasta a reușit se va încerca deplasarea unui pătrat de dimensiune  $3 \cdot N/4$ , altfel se va testa un pătrat de dimensiune  $N/4$  etc. Se va executa algoritmul lui Lee de  $\log N$  ori, ceea ce va conduce la un ordin de complexitate  $O(N^2 \cdot \log N)$ .

În cazul celei de-a treia probleme, căutarea binară se va face în funcție de valorile gradelor de pericolozitate; pentru fiecare alegere a unei astfel de valori se va executa algoritmul lui Dijkstra pentru a determina un drum de lungime minimă, folosind numai muchii cu gradul de pericolozitate mai mic sau egal cu valoarea aleasă. Ordinul de complexitate al acestei soluții este tot  $O(N^2 \cdot \log N)$ .

Cea de-a patra problemă cere alegerea a  $K$  noduri ale arborelui. Dacă am face abstracție de limita pentru numărul de noduri, problema, astfel pusă, ar părea, la prima vedere, NP completă.

Există o variantă abordabilă a acestei probleme, în care se cunoaște distanța cerută  $D$  (cea mai mare distanță față de cea mai apropiată bază) și se cere determinarea numărului minim de baze. Putem rezolva această nouă problemă pornind de la nodurile cu un singur vecin, mergând spre "interiorul" arborelui și amplasând baze în noduri cât mai depărtate de nodurile din exterior, dar totuși suficient de aproape de ele, astfel încât distanța respectivă să nu depășească  $D$ . După construirea arborelui într-o formă ușor de exploatat, acest algoritm ar avea ordinul de complexitate  $O(N)$ . Observăm că, pentru o distanță  $D$  mică, numărul de baze alese va fi mare și reciproc. Distanța minimă cerută este limitată de faptul că trebuie să ne încadrăm în limita de  $K$  baze.

Soluția constă în a alege, pe rând, valori pentru  $D$  folosind căutarea binară. Pentru fiecare dintre aceste valori se va rezolva varianta abordabilă a problemei în  $O(N)$ . Dacă se obțin mai mult de  $K$  baze,  $D$  trebuie mărit; în caz contrar,  $D$  va fi micșorat.

Complexitatea acestei rezolvări este  $O(N \cdot \log N)$ .

Rezolvările pentru problemele a doua, a treia și a patra conduc la următoarea regulă generală: *dacă trebuie obținută o valoare maximă astfel încât să se respecte anumite condiții, și respectarea condițiilor este evidentă pentru valori mai mici, atunci folosind căutarea binară și testarea condiției vom obține rezultatul în  $\log N$  pași*. Regula poate fi adaptată și pentru problemele de minim.

În problemele a doua și a treia valoarea căutată este latura pătratului, respectiv gradul maxim de pericolozitate pentru o muchie. Testarea condițiilor amintite mai sus implică rezolvarea unor subprobleme, folosind algoritmi precum Lee și Dijkstra; aceste subprobleme sunt clasice și, în concluzie, evidente.

Cea de-a patra problemă pare puțin mai complicată, deoarece trecerea la varianta abordabilă a problemei nu este evidentă; folosind principiile din acest articol, astfel de treceri pot fi detectate destul de ușor.

Cea de-a cincea problemă ar putea admite o rezolvare geometrică. Aceasta nu este însă evidentă (cel puțin autorul nu a reușit să o găsească în timp util). Pe de altă parte, se observă că înălțimea la care se ating scândurile crește dacă distanța căutată scade.

Putem reformula cerința problemei astfel: "să se găsească cea mai mare valoare  $D$  astfel încât înălțimea la care se ating scândurile să nu fie mai mică decât  $H$ ". Evident, la limită se obține exact aceeași cerință. Avantajul este că, pentru această abordare, putem folosi căutarea binară în funcție de valoarea  $D$ . Cunoscând  $D$ ,  $L_1$  și  $L_2$ , înălțimea la care se ating scândurile se poate determina destul de ușor, folosind cunoștințe elementare de geometrie plană. Se verifică dacă aceasta este mai mare decât  $H$ ; dacă da,  $D$  va fi crescut, în caz contrar va fi micșorat.

Pare puțin ciudat să rezolvăm o problemă de geometrie folosind căutarea binară a rezultatului, dar practica demonstrează că este foarte eficient.

Abordarea este foarte bună în timpul concursurilor, unde timpul este o resursă critică; implementarea este simplă (fapt foarte important în concurs; *încercați să nu greșiți tocmai căutarea binară!*), timpul de execuție este mic și se economisește timp de gândire.

O problemă care se poate rezolva folosind același principiu este problema **Proc** propusă spre rezolvare la **ONI 2003** la clasele XI-XII.

## Observație

O parte dintre problemele din acest articol admit algoritmi de rezolvare mai eficienți. Am prezentat rezolvarea lor prin metoda din acest articol pentru a evidenția valoarea acesteia; multe probleme aparent dificile devin abordabile, obținându-se algoritmi optimi sau aproape optimi, care, de obicei, se încadrează în timpul de execuție acordat.

## Sugestie

Cititorul este invitat să continue exercițiul, căutând soluții mai eficiente, fără a folosi căutarea binară. Reamintim că, pentru probleme mai dificile, căutarea soluției optime ar putea necesita un timp de gândire important, greu de găsit în timpul concursului.

Mihai Stroe este masterand la Universitatea Politehnica din București și poate fi contactat prin e-mail la adresa [mihai\\_stroe@yahoo.com](mailto:mihai_stroe@yahoo.com).