

Curs Lot Național Juniori

Probleme speciale

Ștefan Dăscălescu

29 Iulie 2023

1 Introducere

La concursurile de informatică, pe lângă problemele obișnuite în care trebuie să citiți inputul (sau eventual îl primiți ca parametri într-o funcție) și trebuie să generați outputul (denumite de regulă probleme batch), mai există și alte tipuri de probleme, pe care le voi numi în cele ce urmează probleme speciale. Acestea sunt de 3 tipuri:

- Probleme interactive: Probleme în care vi se dă un număr limitat de query-uri și trebuie fie să găsiți o anumită valoare/șir, fie să găsiți răspunsul la o cerință mai complexă, răspunsurile la query-uri fiind date de un alt program creat de comisie.
- Probleme output-only: Probleme în care se dau toate datele posibile de intrare și trebuie generate răspunsuri cât mai apropiate de răspunsul optim, de regulă inspirate din probleme NP-complete.
- Probleme de tip two-step (communication tasks): Probleme în care trebuie scrise două programe care pot comunica de regulă între ele doar printr-un output codificat care trebuie generat de primul program pentru a putea ajuta cel de-al doilea program să rezolve problema.

2 Probleme interactive

Când vine vorba de problemele interactive, există foarte multe tipuri de probleme ale căror soluții se folosesc de tot felul de observații în ceea ce privește optimizarea numărului de întrebări, precum și de tehnici care țin de diversele relații și definiții care sunt prezente în probleme.

Există trei tipuri de probleme interactive:

- Probleme în care ni se zice direct complexitatea cerută (e.g. IOI 2014 Rail).
- Probleme în care doar știm numărul maxim de întrebări ce pot fi folosite (e.g. IOI 2013 Cave).
- Probleme care au o limită ascunsă a numărului de întrebări (e.g. IOI 2015 Scales).

Deoarece putem deduce în principiu complexitatea cerută de problemă, de cele mai multe ori nu are sens să încercăm să obținem o complexitate mai bună decât cea de care avem nevoie (dacă numărul de întrebări este n^2 , nu are deloc sens să obținem $\log n$ query-uri).

2.1 Sfaturi practice pentru problemele interactive

În cele ce urmează, voi folosi exemple din [acest material](#), luând de acolo cele mai esențiale sfaturi și problemele la care se regăsesc.

- Există foarte multe probleme interactive care se bazează pe căutări binare sau ternare, exemplele cele mai simple fiind bazate pe aceste tehnici [Guess the Number](#) , [Gradient Descent](#), dar se pot regăsi și exemple mai complicate care țin de diverse optimizări sau de tehnici conexe precum divide et impera, continuitatea funcțiilor (căutarea ternară poate fi folosită pentru aceste probleme).

- Nu ratați relațiile ușor de dedus - De exemplu, pentru problema [Chocolate Bunny](#), o soluție ușor de dedus este aceea de a afla mai întâi maximum, iar mai apoi să aflăm valorile pe rând în $3 \cdot (n - 1)$ query-uri. Totuși, dat fiind că determinăm pe rând valorile, putem reduce numărul de întrebări la $2 \cdot (n - 1)$ query-uri, ceea ce e îndeajuns de bine.
- Încercați să găsiți diferite valori extreme - Putem împerechea diferite tipuri de obiecte folosind stive, cum se poate face la această [problemă](#) dată la un AtCoder Regular Contest. O altă problemă în care putem împerechea diferite tipuri de obiecte este [IOI 2015 Towns](#).
- Câteodată, găsirea unui singur obiect e importantă - În unele probleme interactive, putem elimina pe rând obiecte până ce ajungem la răspuns.
- Rezolvarea cazurilor particulare poate ajuta la o soluție generală
- Probleme în care ne folosim de reprezentările binare pentru a reduce numărul de query-uri. Fie că e vorba de operații pe biți sau de împărțiri precum meet in the middle sau aflarea anumitor submulțimi, aceste tipuri de probleme se pot transpune cu ușurință în probleme interactive.
- Structurile de date pot apărea și ele drept unealtă esențială pentru găsirea soluțiilor la aceste tipuri de probleme, așa cum putem observa în problema [Minimum and Maximum](#) sau dacă vorbim de un exemplu mai dificil, [Ehab and the Big Finale](#)
- De ce nu, putem vorbi și de grafuri mai generale, de multe ori arborii DFS sau arborii parțiali fiind foarte importanți pentru deducerea răspunsurilor cerute de probleme.
- Nu în ultimul rând, și abordările probabilistice se găsesc în aceste tipuri de probleme, mai ales pe Codeforces, exemple putând fi găsite acolo în secțiunile mai dificile ale concursurilor de div2 și div1.

3 Probleme output-only

Deși acest tip de probleme nu apare prea des la concursuri, totuși în ultimii ani au început să apară tot mai des aceste tipuri de probleme, cel mai notabil exemplu fiind IOI 2017 Nowruz. Aceste tipuri de probleme, fiind complet diferite față de celelalte, necesită o abordare specială. Câteva din cele mai utile sfaturi pentru aceste tipuri de probleme sunt

- Aceste probleme nu au în general soluții optime și rapide.
- Trebuie găsit un echilibru între acuratețe și viteza programului.
- Studiați formula de scor și datele de intrare cu atenție, iar pentru cazurile mici, rezolvarea manuală poate fi foarte utilă.
- Obținerea scorurilor parțiale este foarte ușoară și importantă.
- Abordări precum greedy, brute force și algoritmi probabilistici/aproximativi sunt foarte populare pentru aceste probleme.
- E foarte important ca în timp ce lăsați programul să ruleze timp de acele câteva minute bune, să vă gândiți la celelalte probleme pentru a nu pierde timp de implementare și de gândire aiurea.

4 Communication tasks

Pentru aceste probleme, trebuie scrise două programe care de multe ori se bazează pe codificarea/decodificarea unor date de intrare care să permită celor două programe să interacționeze între ele respectând limite stricte în ceea ce privește cantitatea de informație transmisă, deci se impune transmiterea cât mai eficientă pentru al doilea program a informațiilor pe care primul program le primește și le calculează. Deși nu există foarte multe probleme de acest gen, ce se poate remarca la ele este faptul că de multe ori, e bine să începem cu restricții mai relaxate și să ajungem încet-încet la soluție. Totuși, problemele de acest gen tind să fie mai dificile și mai degrabă ceva ce se regăsesc mai des la seniori decât la juniori. Dintre cele mai bune moduri de a aborda problema voi enumera următoarele.

- Folosirea ideilor de la alte probleme rezolvate poate ajuta.
- De multe ori, dacă problema nu e chiar ușoară, obținerea unui punctaj parțial și revenirea la problemă după obținerea punctajelor mai bune la celelalte probleme poate fi o strategie rezonabilă.
- Reprezentările binare, folosirea diverselor operații matematice sau gruparea valorilor în diverse moduri de a le cripta sunt foarte utile la acest tip de probleme.

5 Bibliografie și lectură suplimentară

1. [Probleme interactive](#)
2. [Probleme interactive](#)
3. [Two Way tasks](#)
4. [Probleme interactive si communication tasks](#)
5. [Solutia de la copii3 - infoarena](#)
6. [Ternary Search - Wikipedia](#)
7. [Problemele interactive de pe Codeforces](#)
8. [Interactive, output only Communication Task - HKOI](#)

6 Cuvânt de încheiere

Dacă aveți sugestii, păreri, întrebări sau altceva ce ați vrea să împărtășiți cu privire la articol sau la alte aspecte legate de conținuturile prezentate, mă puteți contacta pe Discord (stefdasca).