TABĂRA DE PREGĂTIRE A LOTULUI NAȚIONAL DE INFORMATICĂ
ORĂȘTIE, 24-29 MAI 2024
BARAJ 3
DESCRIEREA SOLUȚIILOR

COMISIA ȘTIINȚIFICĂ

Gimi p2 drumuri...

Problema 1: Drumuri

Propusă de: prof. Mihai Bunget, Colegiul Național ”Tudor Vladimirescu”, Târgu Jiu

În primul rând vom studia cum se calculează numărul drumurilor pentru un şir A dat, ce reprezintă o permutare a numerelor $1, 2, 3, \ldots, N^2$.

Se reţine mai întâi, pentru fiecare număr de la 1 la N^2, poziţia din şirul A în care se află acesta. Se procesează apoi elementele şirului A în ordine crescătoare, calculând câte drumuri ajung în fiecare element. Elementul de valoare 1 este un punct de start, deci în el ajunge un singur drum. Pentru fiecare element de valoare de la 2 la N^2, numărul drumurilor care ajung în acest element este suma numărului de drumuri care ajung în elementele mai mici decât acesta (care au fost procesate deja) şi din care ne putem deplasa în elementul curent. În final se calculează suma numărului de drumuri care ajung în fiecare element.

În continuare vom arăta că numărul drumurilor pentru orice permutare este cel puţin $2 \cdot N^2 - 2 \cdot N + 1$. Numim predecesor al unui element, un element din şirul A din care ne putem deplasa în elementul curent.

În şirul A există cel puţin un element de start, de exemplu cel cu valoarea 1. Împărţim şirul A în N secvenţe a câte N elemente şi observăm că în fiecare secvenţă $A_{N\cdot(i-1)+1}, A_{N\cdot i+2}, \ldots, A_{N\cdot i}$, unde $1 \leq i \leq N$, sunt posibile $N - 1$ deplasări, câte o deplasare între oricare două elemente alăturate (de la elementul mai mic spre cel mai mare). Deci în fiecare element mai mare, dintre două elemente alăturate, va ajunge un drum venind dinspre elementul mai mic. Dacă elementul mai mic dintr-o pereche este element de start, atunci drumul porneşte din acesta, altfel elementul mai mic are un predecesor (care evident e mai mic decât el) şi continuând inductiv ajungem la un element care e start. Cum există N secvenţe şi în fiecare secvenţă vor fi cel puţin $N - 1$ drumuri, obţinem că vor exista cel puţin $N \cdot (N - 1)$ drumuri care ajung în elemente ale şirului A, venind dinspre elemente alăturate, din aceeaşi secvenţă.

Considerând acum subşirurile de lungime N de forma $A_i, A_{N+i}, A_{2\cdot N+i}, \ldots, A_{(N-1)\cdot N+i}$, unde $1 \leq i \leq N$, observăm că în fiecare subşir se pot face $N - 1$ deplasări între oricare două elemente alăturate, de la elementul mai mic spre cel mai mare. Printr-un raţionament asemănător celui pentru secvenţe se deduce că vor exista cel puţin $N \cdot (N - 1)$ drumuri care ajung în elemente ale şirului, venind dinspre elemente alăturate, din acelaşi subşir.

În total vor fi cel puţin $N \cdot (N - 1) + N \cdot (N - 1) + 1 = 2 \cdot N^2 - 2 \cdot N + 1$ drumuri.

În final trebuie să dăm exemplu de permutare a numerelor $1, 2, 3, \ldots, N^2$ pentru care există exact $2 \cdot N^2 - 2 \cdot N + 1$ drumuri. Pentru a înţelege mai uşor acest exemplu, vom folosi o formă de reprezentare a şirului A mai sugestivă. Astfel, vom împărţi şirul în secvenţe de lungime N, şi scriind aceste secvenţe pe linii diferite vom obţine o matrice pătratică de dimensiuni $N \times N$. Mai exact şirul A se transformă în matricea M astfel încât $M_{ij} = A_{N\cdot(i-1)+j}$, pentru orice $1 \leq i, j \leq N$. Deplasarea dintr-o poziţie k în poziţiile $k - 1, k + 1, k - N, k + N$ în şirul A, unde $k = N \cdot (i - 1) + j$ corespunde deplasării în matricea M din poziţia (i, j) în poziţiile $(i, j - 1), (i, j + 1), (i - 1, j), (i + 1, j)$, respectiv.

Pentru N multiplu de 3 vom bloca cu $*$ anumite poziţii din matricea M, apoi vom completa poziţiile rămase libere cu valori începând de la 1, consecutive, punând fiecare nouă valoare într-o poziţie învecinată cu una deja completată (aceste valori le vom numi valori mici). Blocarea poziţiilor se poate face după următorul algoritm: se pune $*$ pe coloanele 1 şi 3 începând cu linia 2, din doi în doi, apoi pe coloanele 4 şi 6 începem de pe linia 1 din doi în doi ş.a.m.d. Dacă pe coloanele i şi $i+2$ trebuie să punem $*$ pe ultima linie, atunci punem $*$ pe coloana $i+1$ şi ultima linie. De exemplu, pentru $N = 6$, matricea se va completa astfel:

$$\begin{pmatrix} 1 & 2 & 3 & * & 24 & * \\ * & 4 & * & 23 & 22 & 25 \\ 6 & 5 & 7 & * & 21 & * \\ * & 8 & * & 19 & 18 & 20 \\ 10 & 9 & 12 & * & 17 & * \\ 11 & * & 13 & 14 & 15 & 16 \end{pmatrix}$$

Poziţiile marcate cu $*$ se vor completa aleatoriu cu numerele rămase (pe care le vom numi valori mari) şi obţinem:

$$\begin{pmatrix} 1 & 2 & 3 & 26 & 24 & 27 \\ 28 & 4 & 29 & 23 & 22 & 25 \\ 6 & 5 & 7 & 30 & 21 & 31 \\ 32 & 8 & 33 & 19 & 18 & 20 \\ 10 & 9 & 12 & 34 & 17 & 35 \\ 11 & 36 & 13 & 14 & 15 & 16 \end{pmatrix}$$

Pentru a reconstitui şirul A se parcurge matricea pe linii. Observăm că va exista un singur element de start, în fiecare element mic ajunge un singur drum, iar fiecare element mare fiind învecinat doar cu elemente mici, în el va ajunge un singur drum de la fiecare element mic învecinat. Astfel fiecare deplasare orizontală sau verticală posibilă adaugă un singur drum elementului în care se ajunge, obţinând exact numărul minim de drumuri calculat anterior. Pentru N de forma $3 \cdot k + 2$ se completează în mod similar matricea M de dimensiune $3 \cdot k + 3$ apoi se şterg prima linie şi prima coloană, iar pentru N de forma $3 \cdot k + 1$ se completează matricea de dimensiune $3 \cdot k + 3$ şi se şterg primele două linii, prima şi ultima coloană.

Pentru a obţine punctaje parţiale, pentru $N \leq 16$ se poate afişa orice permutare, numărul drumurilor fiind sigur mai mic decât 10^9, însă acest număr trebuie determinat corect. Pentru $N > 16$ se poate alege de exemplu o permutare A pentru care matricea corespunzătoare M să aibă aspectul unei table de şah, punând numerele mici pe pătrăţelele negre iar cele mari pe pătrăţelele albe, obţinând astfel un punctaj acceptabil. De asemenea, numărul drumurilor se poate determina şi recursiv.

Problema 2: GimiGPT

Propusă de: stud. Ioan-Cristian Pop, Universitatea Politehnica București

Plecăm mai întâi de la soluția care obține în jur de 67 de puncte. Ideea acestei soluții este să verificăm, pentru fiecare întrebare, dacă se aseamănă cu oricare dintre întrebările precedente. Dar, pentru a verifica dacă două șiruri q_i și q_j se aseamănă, facem următorii pași:

(1) Dacă diferența de lungime dintre șiruri este cel puțin 2, atunci nu se aseamănă.
(2) Determinăm cel mai lung prefix comun și cel mai lung sufix comun (fie x capătul dreapta al prefixului, și y capătul stânga al sufixului, raportate la întrebarea q_i)
(3) În funcție de valorile lui x și y, avem următoarele cazuri:
   - Dacă $y = 0$, atunci cele două șiruri sunt identice (deci se aseamănă).
   - Dacă $x + 1 < y$, atunci cele două șiruri diferă prin cel puțin două transformări, deci nu se aseamănă.
   - Dacă $|y - x| <= 1$, atunci înseamnă că unul dintre caractere trebuie șters, inserat sau modificat. În funcție de criteriile de asemănare selectate, șirurile se aseamănă sau nu.

Această soluție are o complexitate mare, deoarece verifică șir cu șir. Observăm că prin operațiile de modificare noi putem genera aproximativ 26^2 șiruri care se aseamănă, dar tot vom trece prin lista de N cuvinte.

Optimizarea acestei soluții constă în a genera o parte dintre șirurile care se aseamănă cu q_j odată ce răspundem la întrebare, în mulțimi separate în funcție de tipul de transformare.

Acestea sunt de forma:

- Set original (SO): adăugăm întrebarea q_j.
- Set cu operații de ștergere (SS): adăugăm fiecare variație posibilă (exemplu: din $abbx$ se vor adăuga bbx, abx și $abbx$)
- Set cu operații de modificare (SM): adăugăm fiecare variație posibilă, însă pentru fiecare poziție, plasăm un ”Joker” # în loc să punem toate cele 26 de litere de la a la z (exemplu: din $abbx$ se vor adăuga $\#bbx$, $a\#bx$, $ab\#x$, $abb\#$).

Apoi, pentru fiecare întrebare q_i, se vor face următoarele verificări, în funcție de criteriile de asemănare:

- E: Verificăm dacă există șirul curent (căutăm în setul SO)
- I: Prin inserarea unei litere – verificăm dacă apare șirul curent în setul SS (decât să inserăm în șirul curent, e mai simplu să verificăm prin ștergerea unei litere dacă se poate ajunge în șirul curent).
- S: Prin ștergerea unui singur caracter – pentru fiecare poziție, ”tăiem” litera și verificăm dacă apare șirul transformat prin căutarea în setul SO. Pentru $q_i = abyb$, s-ar putea obține întrebările byb, abb și aby.
- Prin modificarea unei litere – pentru fiecare poziție, modificăm litera în ”Joker” # și verificăm dacă apare șirul transformat prin căutarea în setul SM. Pentru $q_i = abyb$, s-ar putea obține șirurile $\#byb$, $a\#yb$, $ab\#b$, $aby\#$;

Observăm că dacă $q_j = abbx$ iar $q_i = abyb$, cele două nu se aseamănă, pe baza niciunui criteriu de asemănare.

Pentru a verifica dacă există șirul într-un set, pentru 100 de puncte, se pot folosi hash-uri, de exemplu, algoritmul lui Rabin-Karp. Diverse variații ale aceste idei (implementări cu sau fară ”Jokeri”) sunt publicate în arhivă.

Problema 3: p2

Propusă de: Prof. Ionel-Vasile Piț-Rada, Colegiul Naţional “Traian”, Drobeta Turnu Severin

Pentru fiecare valoare $0 \leq N \leq L$ vom calcula prefixele de cel mult 6 cifre ale lui 2^N. Pentru fiecare prefix obținut vom reține lista valorilor N pentru care se obține prefixul respectiv, în ordine crescătoare.

Prefixele se vor calcula folosind operații cu numere mari. Se poate lucra cu baze de reprezentare egale cu diverse puteri ale lui 10.

Pentru interogările de tipul 1 M este suficient să afișăm prima valoare din lista corespunzătoare prefixului M.

Pentru interogările de tipul 2 M N verificăm dacă N apare în lista de valori memorate pentru prefixul M (sau reținem pentru fiecare N prefixul de lungime 6 al lui 2^N și apoi verificăm în cel mult 6 pași dacă M se poate obține ca prefix al lui 2^N).

Pentru interogările de tipul 3 M A B vom căuta binar în lista valorilor pentru care se obține prefixul M pentru a determina numărul de valori incluse în intervalul dat $[A; B]$.

Observații:

- Prefixele pentru puterile 2^N se pot determina și utilizând reprezentarea în virgulă mobilă long double și având grijă doar la păstrarea mantisei, păstrând exponentului o valoare mică.
- Se poate verifica și că pentru a obține prefixul de cel mult 6 cifre este suficient ca în reprezentarea cu numere mari să păstrăm cel mult, aproximativ 20 cifre.

Echipa

Problemele pentru acest baraj au fost pregătite de:

- Prof. Emanuela Cerchez, Colegiul Național ”Emil Racoviță” Iași
- Prof. Adrian Panaete, Colegiul Naţional “August Treboniu Laurian”, Botoşani
- Prof. Ciprian Cheșcă, Liceul Tehnologic ”Grigore C. Moisil”, Buzău
- Prof. Ionel-Vasile Piț-Rada, Colegiul Naţional “Traian”, Drobeta Turnu Severin
- Prof. Mihai Bunget, Colegiul Național ”Tudor Vladimirescu”, Târgu Jiu
- Lector dr. Paul Diac, Facultatea de Informatică, Universitatea “Alexandru Ioan Cuza” Iaşi
- Prof. Dan Pracsiu, Liceul Teoretic “Emil Racoviță”, Vaslui
- Prof. Gheorghe Eugen Nodea, Centrul Județean de Excelență Gorj, Târgu Jiu
- Prof. Daniela Elena Lica, Centrul Județean de Excelență Prahova, Ploiești
- Stud. Ioan-Cristian Pop, Universitatea Politehnica București
- Stud. Dumitru Ilie, Facultatea de Matematică-Informatică, Universitatea București
- Stud. Giulian Buzatu, Facultatea de Matematică-Informatică, Universitatea București
- Stud. Mircea Măierean, Facultatea de Matematică și Informatică, Universitatea ”Babeș-Bolyai”, Cluj-Napoca
- Stud. Bogdan-Ioan Popa, Facultatea de Matematică-Informatică, Universitatea București
- Stud. Andrei Onuț, Yale University, S.U.A.
- Instr. Cristian Frâncu, Nerdvana Education, București
- Prof. Marinel Șerban, Colegiul Național ”Emil Racoviță” Iași
- Prof. Zoltan Szabo, Inspectoratul Școlar Județean Mureș