DESCRIEREA SOLUȚIILOR
LOTUL NAȚIONAL DE INFORMATICĂ
CLUJ, 11-18 MAI 2022
BARAJ JUNIORI 2

EXPRESIA

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

Rezolvarea problemei necesită cunoștințe despre evaluarea unei expresii aritmetice, programare dinamică.

Pentru rezolvarea problemei în cazul general vom nota cu $a[1], a[2], ..., a[m]$ operanzii expresiei și cu $op[1], op[2], ..., op[m-1]$ operatorii expresiei, unde $m = [N/2]+1$.De asemenea, notăm cu $d[i][j][h]$ valoarea maximă a expresiei, ce se obține utilizând operanzii $a[1], a[2], ..., a[i]$, expresie în care s-au folosit j operatori diferiți față de cei corespunzători din expresia inițială, și h operatori de înmulțire.

Valoarea lui $d[i][j][h]$ se calculează dinamic, astfel:

- $d[1][0][0] = a[1]$
- pentru $i \geq 2$, dacă $op[i-1]$ îl vom considera ca fiind ”+”, atunci vom avea
  $d[i][j+w][h] = maxim(d[i][j+w][h], d[i-1][j][h] + a[i])$,
  pentru toate valorile lui j și h pentru care $d[i-1][j][h] \neq 0$, unde $w = 1$ dacă în expresia inițială $op[i-1]$ este ”*” și $w = 0$ în caz contrar. În general, plaja de valori pentru indici este $j = \overline{0, 2 \cdot K}$ și $h = \overline{0, P}$.
- pentru fiecare c de la 1 la P vom considera că ultimii c operatori folosiți sunt ”*”, iar operatorul ce-i precede este ”+”. Astfel, dacă notăm cu $prod$ produsul ultimilor $c+1$ operanzi vom avea $d[i][j+w][h+c] = maxim(d[i][j+w][h+c], d[i-c-1][j][h] + prod)$, pentru toate valorile lui j și h pentru care $d[i-c-1][j][h] \neq 0$. Aici w semnifică numărul operatorilor diferiți față de cei din expresia inițială, dintre ultimii $c+1$ operatori folosiți.
- valoarea maximă a expresiei este maximul valorilor $d[m][j][P]$, unde $j = \overline{0, 2 \cdot K}$ și j număr par.

Pentru cazurile particulare corespunzătoare primelor 4 subtask-uri avem următoarele soluții:

Subtask 1:

Deoarece $K = 0$ trebuie doar să evaluăm expresia. Cât timp avem operator de înmulțire vom calcula produsul operanzilor, iar la întâlnirea operatorului de adunare produsul se va adăuga la suma ce reprezintă în final valoarea expresiei.

Subtask 2:

Aici avem $K = 1$ și $P = 1$, deci unicul operator de înmulțire va trebui să-l așezăm astfel încât să maximizăm valoarea expresiei. Pentru aceasta vom calcula suma tuturor operanzilor, fie aceasta $suma$, iar valoarea maximă a expresiei va fi $maxim(suma + a[i-1] \cdot a[i] - a[i-1] - a[i])$, pentru $i = \overline{2, m}$.

Subtask 3:

Se poate folosi metoda backtracking pentru a genera toate variantele de completare a operatorilor, cu cel mult K inter-schimbări, evaluând de fiecare dată expresia și reținând valoarea maximă.

Subtask 4:

Soluția este asemănătoare cu cea din cazul general, cu mențiunea că nu va mai fi nevoie de indicele j care numără inter-schimbările folosite, deoarece avem $P \leq K$. Astfel vom defini $d[i][h]$ valoarea maximă a expresiei, ce se obține utilizând operanzii $a[1], a[2], ..., a[i]$, expresie în care s-au folosit h operatori de înmulțire. Recurențele se obțin în mod similar.

TUPLECO

Propusă de:
prof. Ciprian Cheșcă, Liceul Tehnologic "Grigore Moisil", Buzău
prof. Ionel Vasile Piț–Rada Colegiul Național "Traian", Drobeta - Turnu Severin

Soluția 1 – backtracking.
Această soluție *"brute force"* testează toate tuplele posibile de câte K numere cu $1 \leq x_1 \leq x_2 \leq ... \leq x_K \leq N$ și determină câte dintre acestea au cel mai mare divizor comun (notat și *cmmdc*) egal cu 1. Ordinul de complexitate este $\mathcal{O}(\mathcal{N}!)$ și soluția obține aproximativ 12 puncte.

Soluția 2 – indicatorul lui Euler.
Pentru $K = 2$ se poate utiliza rezultatul care afirmă că numărul perechilor (a, b) cu $a \leq b \leq M$ și cu proprietatea că a și b sunt prime între ele, este egal cu

$$\phi(1) + \phi(2) + ... + \phi(M)$$

unde $\phi(x)$ reprezintă numărul de numere prime cu x și mai mici decât x și se numește *indicatorul lui Euler*.
Așadar se determină toate valorile lui $\phi(x)$, cu $1 \leq x \leq N$ și apoi se determină suma tuturor valorilor lui $\phi(x)$, cu $1 \leq x \leq N$. Soluția obține 12 puncte.

Soluția 3 – Ciur + Combinări + Funcția Mobius. propusă de prof. Ciprian Cheșcă, Liceul Tehnologic "Grigore Moisil", Buzău

Să observăm pentru început că numărul tuturor vectorilor de numere naturale cu proprietatea

$$1 \leq x_1 \leq x_2 \leq ... \leq x_K \leq N$$

este egal cu numărul vectorilor de numere naturale cu proprietatea

$$1 \leq x[1] < x[2] + 1 < x[3] + 2 < \ldots < x[K-1] + K - 2 < x[K] + K - 1 \leq N + K - 1$$

care este egal cu combinări de $N + K - 1$ luate câte K, notat și $C_{N+K-1}^{K}(1)$
Din acest număr începem să scădem numărul $K - tuplelor$ care au $cmmdc(x_1, x_2, \ldots, x_K) = d$, unde $2 \leq d \leq N$.

- Scădem numărul $K - tuplelor$ care au $cmmdc = 2, 3, 5, \ldots$ și în general pe cele care are un număr impar de factori în descompunerea în factori primi
- Adunăm numărul $K - tuplelor$ care au $cmmdc = 6, 10, 15, \ldots$ și în general pe cele care au un număr par de factori în descompunerea în factori primi.
- Omitem (nu adunăm și nu scădem) numărul $K - tuplelor$ care au $cmmdc$ un număr care nu este liber de pătrate (conțin un factor la o putere mai mare decât 1 în descompunerea în factori primi), cum ar fi $4, 8, 9, 12, \ldots$.

Explicația ar fi următoarea: Din totalul dat de formula de la (1), se scade numărul $K-tuplelor$ care au $cmmdc = 2$, dar nu-l mai scădem pe cel care are $cmmdc = 4, 6, 8, \ldots$ pentru că a fost scăzut odată. Apoi scădem numărul $K - tuplelor$ care au $cmmdc = 3$, doar că numărul $K - tuplelor$ care au $cmmdc = 6$ va fi scăzut de două ori, deci când $cmmdc$ va fi egal cu 6 vom aduna, ș.a.m.d.
Numărul $K - tuplelor$ care au $cmmdc = d$ se poate calcula tot cu ajutorul formulei (1) înlocuind N cu N/d, adică $C_{N/d+K-1}^{K}$ (2)
Să exemplificăm pentru $K = 3$ și $N = 20$:
$C_{N+K-1}^{K} = C_{22}^{3} = 1540$

Scădem:
$C_{N/2+K-1}^{K} = C_{12}^{3} = 220$
$C_{N/3+K-1}^{K} = C_{8}^{3} = 56$
$C_{N/5+K-1}^{K} = C_{6}^{3} = 20$
$C_{N/7+K-1}^{K} = C_{4}^{3} = 4$
$C_{N/11+K-1}^{K} = C_{3}^{3} = 1$
$C_{N/13+K-1}^{K} = C_{3}^{3} = 1$

$C^K_{N/17+K-1} = C^3_3 = 1$
$C^K_{N/19+K-1} = C^3_3 = 1$

Adunăm:
$C^K_{N/6+K-1} = C^3_5 = 10$
$C^K_{N/10+K-1} = C^3_4 = 4$
$C^K_{N/14+K-1} = C^3_3 = 1$
$C^K_{N/15+K-1} = C^3_3 = 1$

Am omis:
$C^K_{N/4+K-1}$
$C^K_{N/8+K-1}$
$C^K_{N/9+K-1}$
$C^K_{N/12+K-1}$
$C^K_{N/16+K-1}$
$C^K_{N/18+K-1}$
$C^K_{N/20+K-1}$

În total obținem $1540 - 220 - 56 - 20 - 4 - 1 - 1 - 1 - 1 + 10 + 4 + 1 + 1 = 1252$ triplete care au $cmmdc = 1$.

Pentru implementare se poate utiliza funcția lui Mobius $\mu(x)$ care este definită astfel:

- $\mu(1) = 1$.
- $\mu(x) = 1$, dacă x conține doar factori primi la puterea întâi și numărul factorilor este par.
- $\mu(x) = 1$, dacă x conține doar factori primi la puterea întâi și numărul factorilor este par.
- $\mu(x) = 0$, dacă x conține cel puțin un factor la o putere mai mare de 1.

O variantă eficientă de implementare a funcție lui Mobius este cu ajutorul unui ciur pentru determinarea numerelor prime iar apoi alt ciur pentru eliminarea numerelor libere de pătrate și stabilirea parității numărului de factori.

În funcție de modul de implementare al funcției lui Mobius această soluție poate obține 80 de puncte. Complexitatea temporală a acestei soluții este $\mathcal{O}(\mathcal{N} \cdot (\log \mathcal{N} + \mathcal{K})$.

Soluția 4. propusă de Piț-Rada Ionel-Vasile – Colegiul Național „Traian", Drobeta Turnu Severin

Notăm cu $f(K,N,d)$ numărul de vectori $x[]$ de lungime K, care conțin numere naturale cu proprietățile: $1 \le x[1] \le x[2] \le \ldots \le x[K] \le N$ și $cmmdc(x[1],x[2],\ldots,x[K]) = d$. Știim de la soluția precedentă că $f(K,N,1) = C^K_{N+K-1}$.

Observăm că $f(K,N,1) + f(K,N,2) + \ldots + f(K,N,N) = C^K_{N+K-1}$. În consecință, putem calcula cerința problemei cu expresia $f(K,N,1) = C^K_{N+K-1} - 1 - (f(K,N,2) + f(K,N,3) + \cdots + f(K,N,N-1))$

Mai putem observa că $f(K,N,i) = f(K,N/i,1)$. Asta înseamnă că rezolvarea problemei noastre $f(K,N,1)$ se reduce la rezolvarea unor probleme asemănătoare, dar de dimensiuni mai mici $f(K,N,1) = C^K_{N+K-1} - (f[K,N/2,1) + f(K,N/3,1) + \ldots + f(K,N/(N-1),1) + f(K,N/N,1))$.

Această idee se poate implementa cu complexitate $\mathcal{O}(\mathcal{N} \times \mathcal{N})$. Putem obține o complexitate mai bună, dacă vom utiliza un algoritm de tip ciur pe un vector definit prin $F[i] = f(K,i,1)$ combinat cu algoritmul *"Difference Array"*, cunoscut și sub numele de "Șmenul lui Mars".

Se inițializează $F[i] = C^K_{i+K-1}$, apoi pentru fiecare $i = 1,2,3,\ldots,N/2$ observăm că $F[i]$ trebuie scăzut din $F[c \cdot i]$, $F[c \cdot i + 1], \ldots, F[c \cdot i + c - 1]$, unde $c = 2,3,\ldots,N/i$. Astfel apare ideea de a utiliza algoritmul *"Difference Array"*: într-un vector $sp[]$ inițializat cu 0 vom scădea $F[i]$ la fiecare poziție $c \cdot i$ și vom crește cu $F[i]$ la fiecare poziție $c \cdot i + c$, dacă nu depășește N. La fiecare pas i, după actualizările anterioare, vom calcula suma parțială $sp[i+1]$ corespunzătoare poziției următoare $i+1$ și calculăm apoi $F[i+1]$. Apoi se continuă cu calcularea valorilor $sp[]$ și $F[]$ până la completarea poziției finale N. Deoarece valorile obținute în calcule pot fi foarte mari vom lucra modulo o valoare mare MOD.

Complexitatea algoritmului este $\mathcal{O}(\mathcal{N} \cdot \log \mathcal{N})$ deoarece atât calcularea valorii C_{N+K-1}^{K} modulo MOD cât și calcularea vectorului $F[]$ modulo MOD se pot realiza în $\mathcal{O}(\mathcal{N} \cdot \log \mathcal{N})$

Soluția 5. propusă de Voroneanu Radu – Google, Zürich

Vom începe de la formula din soluția 4:

$$f(K,N,1) = C_{N-K+1}^{K} - f(K,N/2,1) - f(K,N/3,1) - ... - f(K,N/N,1)$$

Notăm cu A(N) șirul de valori N/2, N/3, ..., N/N. O primă observație este aceea că A(N) conține valori care se repetă. Spre exemplu, pentru $N = 100$, avem că $100/21 = 100/22 = 100/23 = 100/24 = 100/25 = 4$. Astfel, după ce calculăm $f(K,100/20,1)$, nu mai este necesar să recalculăm și pe $f(K,100/21,1)$ ș.a.m.d.

Setul de valori unice din $A(N)$, notat cu $U(N)$, este $\{N/2, N/3, ..., N/\sqrt{N}, \sqrt{N}, \sqrt{N}-1, \ldots, 1\}$ (mică atenție când N este pătrat perfect). De altfel, o valoare T din $U(N)$ o să apară în $A(N)$ de $[N/T] - [N/(T+1)]$ (unde $[x]$ este partea întreagă a lui x). În acest context, $[N/T]$ reprezintă cea mai mare valoare i astfel încă $[N/i] = T$. Putem acum rescrie formula anterioară:

$$f(K,N,1) = C_{N-K+1}^{K} - \sum_{T \in U(N)} \left([N/T] - [N/(T+1)] \right) * f(K,T,1)$$

Astfel, dacă am calculat deja toate $f(N,T,1)$ pentru $T \in U(N)$, putem calcula $f(K,N,1)$ în $O(\sqrt{N})$. Rămâne acum de calculat valorile $f(N,T,1)$ într-un mod similar.

Pentru a face o a doua observație importantă, vom porni de la un exemplu: $N = 20$. Atunci $f(K,20,1)$ depinde de $f(K,10,1), f(K,6,1), f(K,5,1), f(K,4,1), f(K,3,1), f(K,2,1), f(K,1,1)$. Pentru a calcula pe $f(K,10,1)$, acesta o să depindă la rândul sau de $f(K,5,1), f(K,3,1), f(K,2,1), f(K,1,1)$. Observația acum este că toate valorile de care depinde $f(K,10,1)$ erau prezente și în valorile de care depinde $f(K,20,1)$. Acest lucru se întâmplă deoarece împărțind N printr-un număr X și apoi împărțind rezultatul la un alt număr Y, este echivalent cu împărțirea lui N la $(X \cdot Y)$. Astfel, algoritmul va itera prin valorile lui $U(N)$ în ordine crescătoare și va aplica formula anterior discutată. Valorile necesare la un anumit pas vor fi deja calculate în pașii anteriori.

Atenție la calculul combinărilor. Notăm cu $f[i] = i!\%MOD$ și $inv[i]$ =inversul modular a lui $f[i]$ (pentru orice $i < MOD$). Calculul lui $f[]$ se face începând cu $f[0] = 1$ și apoi $f[i] = (f[i-1] \cdot i)\%MOD$. Pentru calculul lui $inv[]$, se calculează mai întâi $inv[MOD-1]$ folosind spre exemplu Euclid extins. Apoi se folosește formula de recurență $inv[i] = (inv[i+1] \cdot (i+1))\%MOD$. Acestea două pot fi folosite pentru calculul combinărilor. Un caz special apare când trebuiesc calculate factoriale și inverse modulare a valorilor mai mari ca MOD. Pentru o astfel de valoare A, conform restricțiilor problemei, se observă că sunt maxim 6 numere divizibile cu MOD în $1, 2, \cdot, A$. Se poate atunci împărți $A!$ ca produs de 6 secvențe de înmulțiri.

Vaslui1475

Propusă de: prof. Marinel Șerban, Colegiul Național "Emil Racoviță" Iași

Soluția 1 – 100p. propusă de Piț-Rada Mihai-Cosmin – Bolt

Dreptele de interes sunt fie paralele cu diagonala principală, fie paralele cu diagonala secundară. Ecuaţiile acestor drepte arată astfel:

- $x - y = constant$, cu valori în intervalul $[1-N, N-M]$.
- $x + y = constant$, cu valori în intervalul $[2, N+M]$.

Pentru prima cerinţă, trebuie determinată o astfel de dreaptă (având una dintre cele două forme) astfel încât va separa cele P puncte în două zone conţinând fiecare jumătate. Cu ajutorul unei mapări a constantei ecuaţiei pe un vector se poate contoriza câte puncte se află pe fiecare diagonală. Cu ajutorul unei sume parţiale se detectează o diagonală al cărei contor este 0, iar înaintea acesteia s-au numărat jumătate dintre punctele iniţiale. Evident P trebuie să fie par pentru a exista soluţie.

Cerinţa a doua se rezolvă similar, detectând câte o dreaptă de fiecare tip. Acest lucru nu este suficient pentru a garanta o împărţire corectă a punctelor, astfel o nouă scanare a listei de puncte este necesară pentru a număra câte puncte se află în fiecare zonă (sub prima dreaptă și sub a doua dreaptă, sub prima dreaptă și deasupra celei de-a doua drepte etc.). Soluţia va exista doar dacă P este divizibil cu 4. De asemenea, trebuie tratat cazul special când $P = 0$ asigurând că dreptele se intersectează și niciuna dintre zone nu este vidă.

Soluția 2 – 100p. propusă de Voroneanu Radu – Google, Zürich

Se pornește de la Soluția 1 în care diagonalele au fie suma fie diferența constantă. Fie $x[i]$, $y[i]$ coordonatele soldatului i. Vom construi doi vectori $s[i] = x[i] + y[i]$ și $d[i] = x[i] - y[i]$. Se observa ca pentru ca o diagonala sa împartă punctele în doua părți egale, valoarea diagonalei trebuie sa împartă valorile vectorului s (sau d în funcție de caz) în doua jumătăți egale. Valoarea care împarte valorile unui vector în doua jumătăți egale este de fapt medianul vectorului. Atenție deoarece numărul de soldați trebuie sa fie par pentru existenta unei soluții, ceea ce înseamna ca soluția căutată este o valoare intre cele 2 mediane (exclusiv, deoarece nu dorim ca diagonala sa intersecteze soldații). Astfel, trebuie determinat medianul lui s sau d. Acesta poate fi calculat cu mai multe metode - o sortare directa a elementelor; algoritmul de calcul al celui de al n-lea element al unui vector, bazat pe ideea de sortare "*QuickSort*"; căutare binara a valorii medianului.

Soluția 3 – 60-70p. propusă de Șerban Marinel – Colegiul Național ,,Emil Racoviță", Iași

(Pentru valori mari ale lui N și M această soluție nu intră în timp)

- citesc datele, memorând tabloul (*memorie insuficientă pentru valori mari ale lui N și M*)
- dacă P nu este par nu am soluție nici la a) nici la b)
- dacă P nu este divizibil cu 4 aș putea avea soluție la a)
- verific diagonale de tip '/' diagonale paralele cu diagonala secundară /
  - determin diagonalele și le rețin în vectorul $d1$ împreună cu numărul de luptători de deasupra ei
  - verific dacă diagonala / $(i,c1)$-$(l1,j)$ conține sau nu luptători
- verific diagonale paralele cu diagonala principală
  - determin diagonalele și le rețin în vectorul $d2$ împreună cu numărul de luptători de deasupra ei
  - verific dacă diagonala $(i,c1)$-$(l1,j)$ conține sau nu luptători
- analizez separat vectorii $d1$ și $d2$ pentru punctul a) și împreună pentru punctul b); la determinarea unei/unor diagonale corecte afișez și opresc procesul

Echipa

Problemele pentru această etapă au fost pregătite de:

- Nicoli Marius – Colegiul Național ,,Frații Buzești", Syncro Soft, Craiova
- Boian Flavius – Colegiul Național ,,Spiru Haret", Târgu-Jiu
- Bunget Mihai – Colegiul Național ,,Tudor Vladimirescu", Târgu-Jiu
- Cerchez Emanuela – Colegiul Național ,,Emil Racoviță", Iași
- Cheșcă Ciprian – Liceul Tehnologic ,,Grigore Mosil", Buzău
- Frâncu Cristian – Clubul Nerdvana București
- Lica Daniela – Centrul Județean de Excelență Prahova, Ploiești
- Manz Victor – Colegiul Național de Informatică ,,Tudor Vianu", București
- Nodea Gheorghe-Eugen – Colegiul Național ,,Tudor Vladimirescu", Târgu-Jiu
- Oprița Petru – Liceul ,,Regina Maria", Dorohoi
- Pintea Adrian – Inspectoratul Școlar Județean Cluj
- Piț-Rada Ionel-Vasile – Colegiul Național ,,Traian", Drobeta Turnu Severin
- Piț-Rada Mihai-Cosmin – Bolt
- Pop Ioan Cristian – Universitatea Politehnica București
- Pracsiu Dan – Liceul Teoretic ,,Emil Racoviță", Vaslui
- Șerban Marinel – Colegiul Național ,,Emil Racoviță", Iași
- Tulbă-Lecu Theodor-Gabriel – Universitatea Politehnica București
- Voroneanu Radu – Google, Zürich