DESCRIERE SOLUŢII

PROBLEMA 1 PITICI

Autor: elev Nitu Vlad
Colegiul Național "Mihai Viteazul", Ploiești

Soluție 50 puncte (complexitate : O(N*Q))

Aceasta este soluția brută. Folosind vectorul citit, pentru fiecare întrebare primita, vom parcurge vectorul între pozițiile date (stânga și dreapta , inclusiv) și vom calcula suma dintre aceste poziții.Vom avea grija că variabila pentru suma să fie declarata de tipul "long long" .

Soluție 100 puncte (Complexitate : O(N))

v[i] = suma elementelor de la 1 la i (inclusiv)

Vom construi vectorul "v" în timp ce citim elementele, prin relația de recurenta:

→ v[i] = v[i-1] + h; h=înălțimea piticului "i".

Pentru fiecare operație , pentru a calcula suma dintre stânga şi dreapta, avem:

sol = v[dr] – v[st-1] , ceea ce reprezintă exact suma elementelor începând cu poziția 'stânga' , pana la 'dreapta' (inclusiv) , aceasta operație fiind realizata in O(1).

PROBLEMA 2 JOACA

Autor: prof. Lincan Iulia
Colegiul Național "Mihai Viteazul", Ploieşti
Soluție: Drăgan Andrei - Colegiul Național "Mihai Viteazul", Ploieşti

Pentru fiecare celula ocupată, vom genera un "cod" al acesteia în funcție de linia și coloana pe care o are. Astfel, vom considera că o celula ce se afla pe linia x și coloana y, va avea codul = m*(x-1) + y, cod ce reprezintă poziția celulei respective, daca am înșira liniar toate celulele de pe fiecare linie, în ordinea coloanelor. (de exemplu, celula (3,5) vă fi într-o matrice 10x10 codificata cu valoarea 2*10+5=25). Fiecare cod generat din datele de citire, vă fi marcat cu 1 într-un vector de tip "bool".

De asemenea, vom reţine următorii vectori:

- linie[i] = 1/0 dacă am sau nu am cel puțin un loc de joacă pe linia i;
- col[j] = 1/0 dacă am sau nu am cel puțin un loc de joacă pe coloana j;
- sol_linie[i] = 1/0 dacă este nevoie sau nu să păstrez aleea reprezentată de linia i;
- sol_col[j] = 1/0 dacă este nevoie sau nu să păstrez aleea reprezentată de coloana j;
-

Vom parcurge fiecare linie cu o structură repetitivă de la 1 la n, iar dacă la o poziție K:

linie[K] = 0 și linie[K-1] != 0, înseamnă că aleea de pe linia K, va reprezenta aleea de lățime 1, deci vom marca sol_linie[K] = 1; (aceasta alee fiind prima dintr-un posibil şir de mai multe alei orizontale de pe linii consecutive).

linie[K] = 0 și linie[K-1] = 0, înseamnă că vom marca sol_linie[K]=0, pentru că nu ne mai interesează aleea de pe linia K , deoarece deja avem aleea de pe linia K-1.

Linie[K] != 0, clar punem sol_linie[K]=1, pentru că linia K nu mai reprezintă o alee.

Analog vom parcurge și pentru coloane și vom calcula sol_col.

Atenție: se initializeaza linie[0] = -1, col[0] = -1, în cazul în care linie[1] = 0 (atunci aceasta este bună, ea reprezentând un “start” pentru un posibil şir de alei consecutive pe linii, deci trebuie să o păstram) – același motiv și pentru coloane.

În final, vom afișă doar liniile și coloanele care au sol_linie, respectiv sol_col diferite de 0, iar pentru acestea, vom recoda toate elementele și vom afișa 1, sau 0 în funcție de vectorul “bool” format la început.

PROBLEMA 3 VALUTA

Autor: elev Iancu Andrei
Colegiul Național “Mihai Viteazul”, Ploieşti

Pentru a putea transporta toți banii conform regulilor, numărul de bancnote de fiecare tip trebuie sa se împartă exact la capacitatea cutiei alese, fără a fi amestecate mai multe tipuri de bancnote. De asemenea, Budi trebuie sa aleagă cele mai mari cutii pentru a fii cât mai econom. Așa că pentru fiecare set de date vom parcurge vectorul de cutii, care va fi sortat descrescător la început, iar pentru prima cutie care verifica cerința problemei, vom afişa şi soluția. Dacă Budi nu poate împarți banii in niciun tip de cutie, atunci se va afișa “imposibil”.

PROBLEMA 4 ELBUDI

Autor: elev Cîrstea Andrei
Colegiul Național “Mihai Viteazul”, Ploieşti

Se va prelucra caracter cu caracter fiecare ecuație determinând în fiecare membru suma termenilor cu x și a termenilor liberi (fară x). Dacă numărul de x în membrii este egal atunci vom avea o ecuație imposibilă (suma_termeni_stanga ! = suma_termeni_dreapta) sau o ecuație nedeterminată (suma_termeni_stanga = suma_termeni_dreapta) și vom creste numărul de ecuații cu efortul rezolvării 0. Daca numărul de x diferă în cei doi membrii, atunci vom calcula în modul diferența lor (vom scădea mereu pe cel mic în cel mare) și valoarea diferenței dintre sumele termenilor liberi în partea cealaltă a ecuației (rezolvarea matematică a unei ecuații cu o singură necunoscută). Dacă valoarea soluției este 0 atunci vom creşte numărul de ecuații cu efortul rezolvării 0. Altfel vom introduce valoarea soluției într-un vector și vom creste numărul soluțiilor mai mari ca 0. După ce am prelucrat toate ecuațiile, vom sorta crescător vectorul soluțiilor, îl vom parcurge și vom adauga câtul împărțirii soluției cu indexul i la i + nr_ecuatii_cu_efort_egal_cu_0 (ecuațiile cu soluții mari vor fi lucrate primele, adică împărțite cu cele mai mari numere, iar cele cu soluții mici vor fi lucrate ultimele, împărțite cu cele mai mici numere).