Problema Concert

Clasa a IX-a
Autor Mrd. Răzvan-Daniel Zoltan, Universitatea Babeș-Bolyai, Cluj-Napoca

Formalizarea problemei

Având o matrice binară A de dimensiuni NxM, avem T operații de actualizare în care, pentru submatricea definită prin colțul stânga-sus de coordonate (x_1, y_1) și, respectiv, prin colțul dreapta-jos de coordonate (x_2, y_2), elementele își vor schimba valoarea din 1 în 0 și vice-versa (cu alte cuvinte, după operația de actualizare, valoarea lor va fi XOR-ată cu 1).

După executarea tuturor operațiilor de actualizare, vom avea Q întrebări de forma: „Care este valoarea elementului A_{x_Q,y_Q}?“

Abordarea 1 - complexitate $\mathcal{O}(T \cdot N \cdot M + Q)$

Pentru valori mici ale lui N și M, putem implementa o soluție brută în care vom actualiza secvențial fiecare valoare din submatricea definită prin colțul stânga-sus de coordonate (x_1, y_1) și, respectiv, prin colțul dreapta-jos de coordonate (x_2, y_2) pentru fiecare coordonată.

Această soluție va obține în jur de 20 de puncte.

Abordarea 2 - complexitate $\mathcal{O}(T \cdot N \cdot M \cdot \frac{1}{W} + Q)$

Pentru valori ceva mai mari ale lui N și M, putem face următorul artificiu: vom împărți coloanele matricei în calupuri (chunk-uri) de 32 sau 64 de biți, folosindu-ne de reprezentarea internă a numerelor de tip **int** sau **long long** (sau variantele lor **unsigned**).

În acest sens, o operație de actualizare va avea complexitatea $\mathcal{O}(N \cdot M \cdot \frac{1}{W})$, unde W ia valoarea $\frac{1}{32}$ sau $\frac{1}{64}$, în funcție de implementare. Motivul pentru care această soluție se comportă mai bine în practică decât cea de la prima abordare este că pentru fiecare linie din submatricea de actualizare vom efectua operații pe biți în timp constant ($\mathcal{O}(1)$) pentru fiecare calup relevant, obținând o constantă mai bună care se comportă mai bine în practică.

Această soluție va obține în jur de 40-50 de puncte, în funcție de implementare.

Abordarea 3 - complexitate $\mathcal{O}(N \cdot M + T + Q)$

Pentru obținerea punctajului maxim, putem merge pe următoarea idee: definim o matrice auxiliară B care pentru fiecare operație de actualizare va suferi următoarele modificări:

$$\begin{cases} B_{x_1,y_1} = B_{x_1,y_1} \oplus 1 \\ B_{x_2+1,y_1} = B_{x_2+1,y_1} \oplus 1 \\ B_{x_1,y_2+1} = B_{x_1,y_2+1} \oplus 1 \\ B_{x_2+1,y_2+1} = B_{x_2+1,y_2+1} \oplus 1 \end{cases} \tag{1}$$

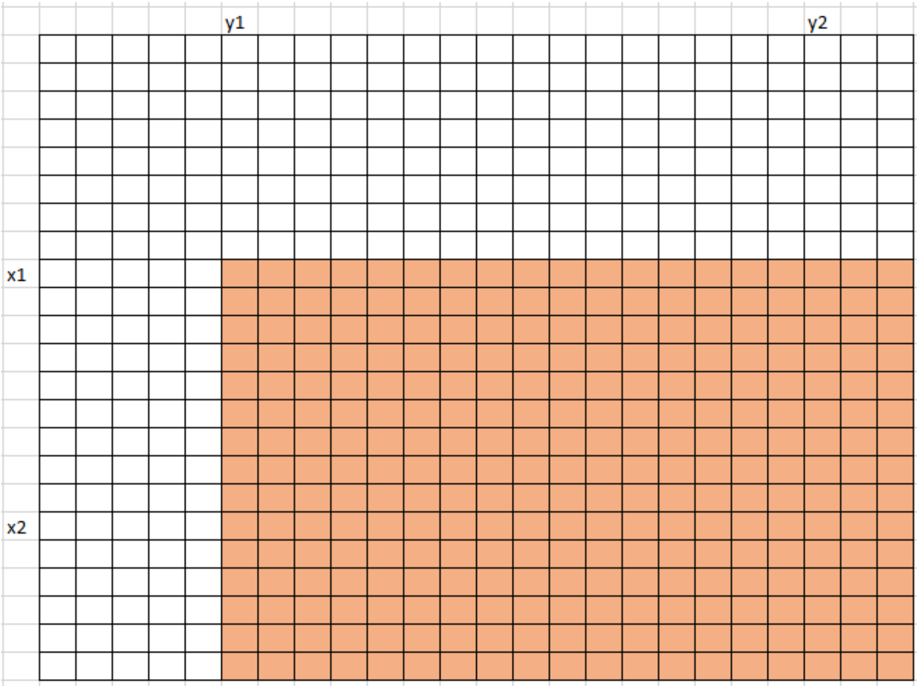
În setul de actualizări de mai sus pe matricea B, operatorul de mai sus reprezintă operatorul XOR (sau exclusiv pe biți).

Apoi, vom efectua XOR-uri parțiale pe matrice (foarte similar cu generarea sumelor parțiale pe matrice, dar ajustând recurența în următorul fel):

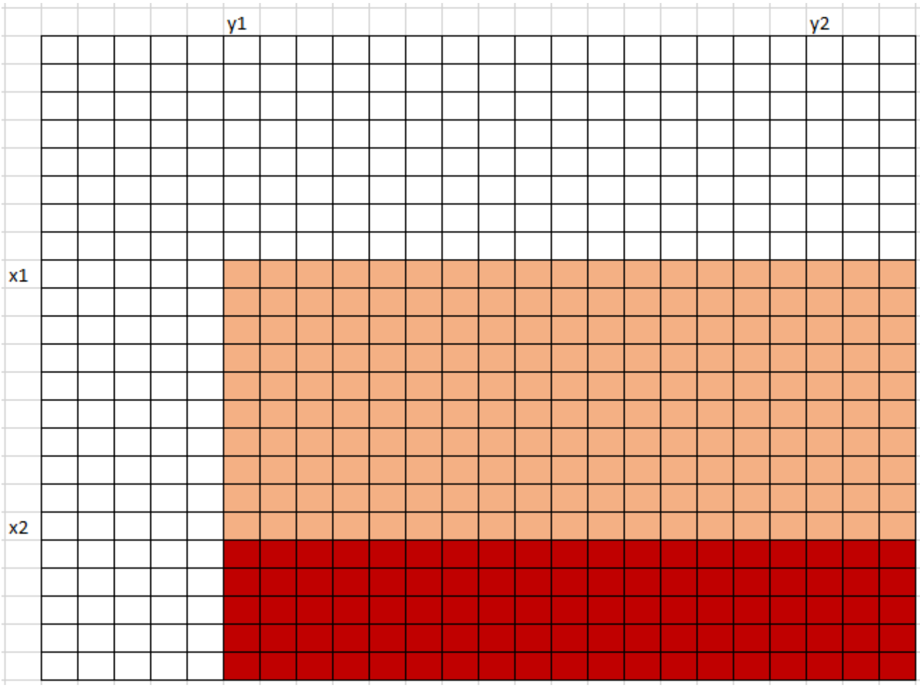
$$B_{i,j} = B_{i-1,j} \oplus B_{i,j-1} \oplus B_{i-1,j-1} \tag{2}$$

În final, răspunsul la fiecare întrebare va fi $A_{x_Q,y_Q} \oplus B_{x_Q,y_Q}$. Utilitatea matricei B este efectuarea operațiilor de actualizare în timp constant, motivul pentru care această abordare funcționează fiind descris mai jos.

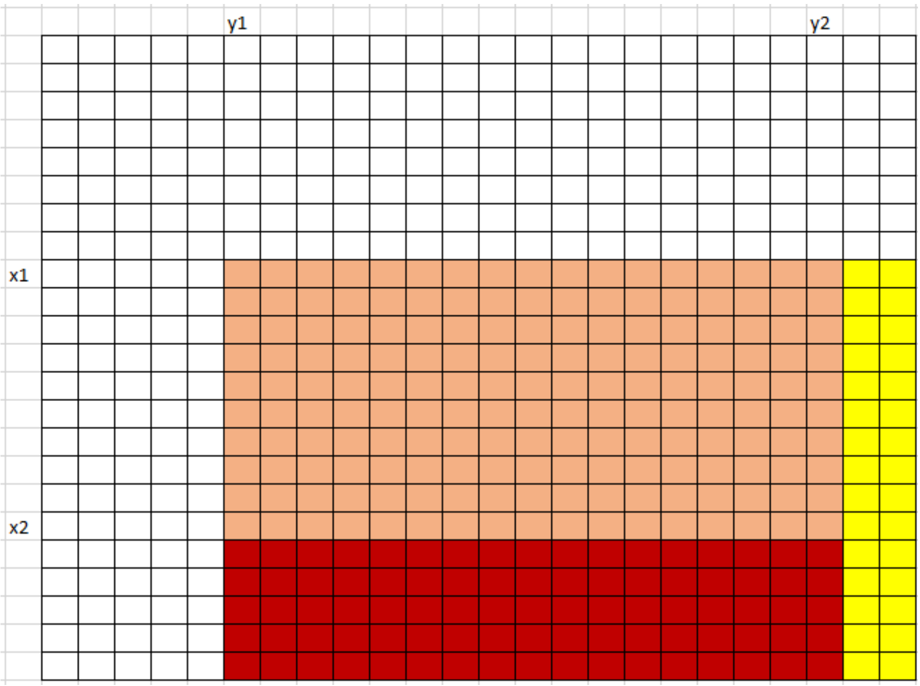
În submatricea de actualizare definită prin colțul stânga-sus de coordonate (x_1, y_1) și, respectiv, prin colțul dreapta-jos de coordonate (x_2, y_2), prima operație de actualizare ($B_{x_1,y_1} = B_{x_1,y_1} \oplus 1$) va propaga în toată submatricea de colț stânga sus (și colț dreapta-jos la sfârșitul matricei inițiale) operația de XOR cu 1 după efectuarea sumei parțiale, efectul fiind colorat cu oranj în imaginea de mai jos.



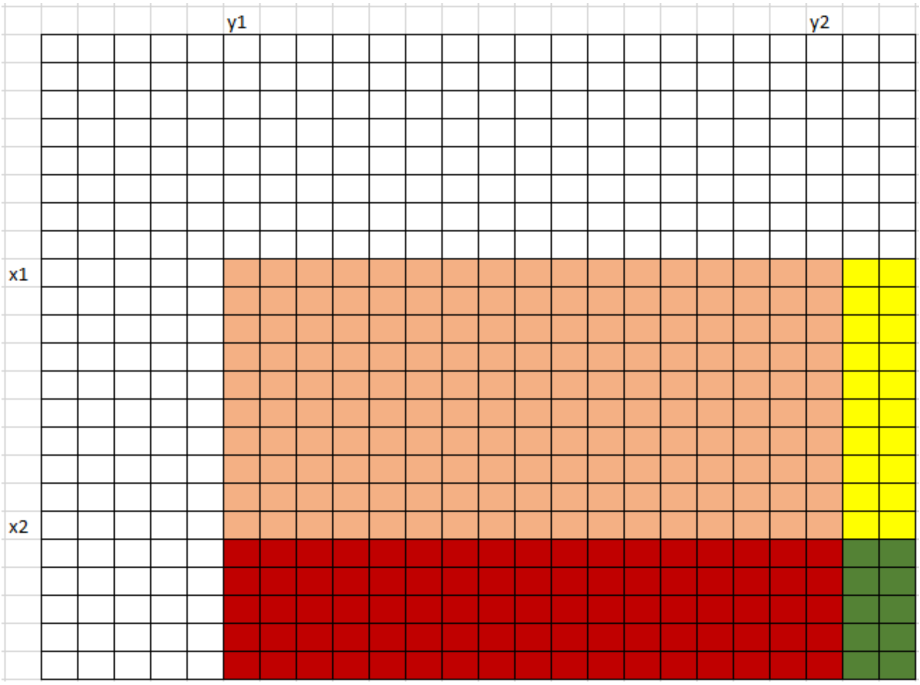
Cea de-a doua operație ($B_{x_2+1,y_1} = B_{x_2+1,y_1} \oplus 1$) care reprezintă operația inversă din prima operație (în cazul acesta este aceeași) din cele patru va opri propagarea pe liniile mai mari decât x_2, scopul nostru final fiind să avem efectul primei operații restrâns între colțul stânga-sus de coordonate (x_1, y_1) și, respectiv, prin colțul dreapta-jos de coordonate (x_2, y_2). Efectul acesteia este colorat cu roșu în imaginea de mai jos.



Cea de-a treia operație ($B_{x_1,y_2+1} = B_{x_1,y_2+1} \oplus 1$) care reprezintă tot operația inversă din prima operație din cele patru va opri propagarea pe coloanele mai mari decât y_2. Efectul acesteia este colorat cu galben în imaginea de mai jos.



După cea de-a treia operație observăm că în submatricea cu colțul stânga-sus de coordonate (x_2+1, y_2+1) și, respectiv, prin colțul dreapta-jos de coordonate (N, M) propagarea de la prima operație a fost oprită de două ori așa că cea de-a patra operație $(B_{x_2+1,y_2+1} = B_{x_2+1,y_2+1} \oplus 1)$ este una de neutralizare în care readucem la normal submatricea definită anterior. Efectul acesteia este colorat cu verde în imaginea de mai jos.



După cele patru operații, actualizarea este restrânsă la submatricea de actualizare dorită inițial. Cele patru operații au fost făcute în timp constant, toate operațiile de actualizare la un loc combinându-se și determinând corect matricea finală după efectuarea XOR-urilor parțiale.

Abordarea 4 - complexitate $\mathcal{O}(N \cdot M + T + Q)$

Tot pentru obținerea punctajului maxim, pe aceeași idee, folosind setul de actualizări în matricea B definit mai jos, putem număra de câte ori a fost actualizat o anumită celulă din matrice. Dacă aceasta a fost actualizată de un număr par de ori, aceasta nu-și va schimba valoarea față de cea inițială. În caz contrar, aceasta va deveni 1 din 0 și vice-versa.

$$\begin{cases} B_{x_1,y_1} = B_{x_1,y_1} + 1 \\ B_{x_2+1,y_1} = B_{x_2+1,y_1} - 1 \\ B_{x_1,y_2+1} = B_{x_1,y_2+1} - 1 \\ B_{x_2+1,y_2+1} = B_{x_2+1,y_2+1} + 1 \end{cases} \tag{3}$$

Notă

În argoul competițiilor de informatică, metoda aplicată în abordările 3 și 4 este cunoscută sub numele de „Șmenul lui Mars“, aplicat în varianta bidimensională.