Editorial Neoficial ONI 2013
Baraj Juniori

Stefan Dascalescu

Cuprins

1 Latin

Autor Solutie: Stefan Dascalescu

1.1 Soluție brută

O primă soluție ar consta în a verifica pentru fiecare colț din stânga sus și fiecare lungime a pătratului dacă este latin sau nu. O soluție care rulează în $O(n^4)$ constă în a fixa colțul din stânga sus și colțul din dreapta jos și pe măsură ce adăugăm puncte noi, putem verifica în $O(n)$ dacă noul pătrat obținut este latin pe baza pătratului anterior. Complexitatea este dată de faptul că avem n^2 posibilități de a alege colțul din stânga sus, n lungimi diferite și la fiecare pas, extindem pătratul cu aproximativ n poziții.

1.2 Soluție optimă

Faptul că pătratul inițial este latin ne garantează faptul că pentru oricare linie și coloană din matrice, toate pătratele sunt distincte și asta ne permite să verificăm cu ușurință dacă un pătrat este latin. Astfel, noi știm că pentru toate valorile x, avem un singur x pe fiecare linie și coloană, deci asta înseamnă că suma minimă a unei fâșii de lungime k (deci $1 \times k$ sau $k \times 1$ este $\frac{k \cdot k+1}{2}$, iar suma totală minimă este deci $k \times \frac{k \cdot k+1}{2}$.

Pentru a putea afla cu ușurință valorile acestor sume pentru fiecare alegere de a fixa colțul din stânga sus și o lungime, putem folosi sumele parțiale și verificăm dacă suma obținută este egală cu valoarea minimă a unei astfel de sume parțiale. Astfel, complexitatea finală se reduce la $O(n^3)$, ceea ce este îndeajuns pentru problema dată.

1.3 Soluție bonus

Dat fiind că avem valori de la 1 la n, se poate obține o soluție și în complexitate $O(n^2)$ folosind păduri de mulțimi disjuncte pentru a găsi potențialele pătrate latine. Pe scurt, pentru fiecare număr de la 1 la n, vom reuni pătratele egale cu x la mulțimile existente și vom verifica pentru fiecare mulțime dacă are exact i^2 numere și valorile sunt răspândite pe o suprafață de i linii și i coloane, soluția fiind corectă datorită presupunerii că pătratul este latin.
Detaliile de implementare se lasă ca exercițiu pentru cititor.

1.3.1 Cod sursă

Soluția lui Ștefan

2 Swap

Autor Solutie: Stefan Dascalescu

2.1 Soluție

Pentru a rezolva această problemă, vom pleca de la distribuția inițială a perechilor de paranteze, aflând pentru fiecare paranteză deschisă unde se va închide, obținând astfel un cost inițial. Pentru a afla acest cost, vom ține cu ajutorul unei stive pozițiile unde avem paranteze deschise și de fiecare dată când avem o paranteză închisă, vom scoate cea mai apropiată paranteză deschisă din stivă și vom actualiza costul cu valoarea distanței dintre acele două poziții.
Acum, vrem să aflăm pentru cerințele 2 și 3 costul minim pentru a obține o parantezare corectă conform algoritmului folosit la cerința 1. Mai întâi, nu are niciun sens să dăm swap la două paranteze identice deoarece răspunsurile ar rămâne identice, deci ne mai rămân două cazuri.
Indiferent de caz, trebuie să ne asigurăm că dacă facem acea interschimbare, parantezarea rămâne corectă, lucru ce se poate verifica destul de ușor folosind un vector de sume parțiale în care ps_i reprezintă diferența dintre numărul de paranteze deschise și numărul de paranteze închise până la poziția i. Vom presupune în ambele cazuri că schimbăm parantezele de la pozițiile i și $i + 1$, iar după fiecare încercare, verificăm dacă costul scade sau nu.

2.1.1 Cazul ()

În acest caz, dacă $ps_{i-1} > 0$, parantezarea rămâne validă și paranteza închisă se deplasează la stânga cu o poziție, iar paranteza deschisă la dreapta cu o poziție, deci costul scade cu 2.

2.1.2 Cazul)(

În acest caz, parantezarea rămâne clar validă și paranteza închisă se deplasează la dreapta cu o poziție, iar paranteza deschisă la stânga cu o poziție, deci costul crește cu 2.

2.1.3 Cod sursă

Soluția lui Ștefan

3 Zmax

Autor Solutie: Stefan Dascalescu

3.1 Soluție

Pentru a rezolva această problemă, vom vrea să precalculăm mai întâi pentru fiecare pătrat din matrice subsecvența de sumă maximă de pe aceeași linie care începe sau se termină la poziția (i, j) și are lungimea cel puțin 2. Acest lucru se poate precalcula în diverse moduri, folosind unul din algoritmii cunoscuți pentru aflarea subsecvenței de sumă maximă.

Apoi, o soluție inițială ar consta în fixarea celor două pătrate ce vor forma partea diagonală a Z-ului și folosirea sumelor parțiale calculate anterior pentru a afla un potențial răspuns maxim, soluție ce rulează în $O(n^4)$.

O primă optimizare, care va fi și suficientă pentru obținerea punctajului maxim în versiunea actuală a problemei constă în a observa faptul că sunt cel mult $2 \times n$ diagonale de care ne interesează, iar pentru fiecare diagonală, avem cel mult n puncte din care putem alege capetele părții diagonale a Z-ului. Folosindu-ne de sumele parțiale calculate anterior, putem afla răspunsul în $O(n^3)$, ceea ce va lua 100.

Această soluție se poate optimiza mai departe folosind o dinamică pe care o putem calcula pe baza precalculărilor inițiale. Putem defini $dp[i][j]$ ca fiind suma maximă a unui 7 (întreg Z-ul cu excepția ultimei părți) care se termină pe poziția (i, j). Pentru a simplifica calculele inițiale, vom presupune că putem avea și 7-uri degenerate (fără partea din mijloc, practic o linie). Astfel, $dp[i][j]$ va fi maximul dintre $dp[i-1][j+1] + mat[i][j]$ și $sum[0][i][j]$, unde $sum[0][i][j]$ este subsecvența de sumă maximă care se termină pe poziția (i, j) iar $mat[i][j]$ este pătratul de pe poziția (i, j). Acum, tot ce ne rămâne de făcut pentru o poziție oarecare (x, y) este să ne folosim de dinamica anterior pentru a afla răspunsul maxim, care e maximul de forma $sum[1][i][j] + dp[i-2][j+2] + mat[i-1][j+1]$, unde $sum[1][i][j]$ este subsecvența de sumă maximă care începe pe poziția (i, j). Această soluție are complexitate $O(n \times m)$ ceea ce va obține 100 și este și soluția optimă pentru această problemă.

3.1.1 Cod sursă

Soluția patratica a lui Ștefan