Probleme SDA

Dreptunghiuri5

Înțelegerea problemei:
Scopul este de a determina numărul dreptunghiurilor formate din valorile „0” dintr-o matrice dată.

Structura codului:
Gestionarea intrărilor:
Citește dimensiunile matricei (rânduri și coloane).
Citește valorile matricei din fișierul de intrare.

Funcția setBorder:
Setează marginea matricei cu 1s.

Functie principala:
Inițializează o matrice booleană de dimensiune maximă (1005x1005).
Setează chenarul matricei la 1s folosind funcția setBorder.
Citește valorile matricei din fișierul de intrare.
Utilizează o stivă pentru a procesa matricea rând cu rând și pentru a găsi dreptunghiuri.
Utilizează un heightStack pentru a urmări „0” consecutive în fiecare coloană.
Iterează prin fiecare rând, actualizând informațiile de înălțime pentru fiecare coloană.
Urmărește lățimea și înălțimea dreptunghiurilor posibile folosind o abordare bazată pe stivă.
Numărează numărul de dreptunghiuri formate din valorile „0”.
Afișează numărul dreptunghiurilor găsite.

Observatii:
Procesarea matricei:
Codul iterează prin matrice rând cu rând, utilizând o stivă pentru a calcula numărul dreptunghiurilor formate din „0” consecutive.
Menține o stivă (heightStack) pentru a urmări înălțimile și lățimile pentru identificarea dreptunghiurilor.

Identificare dreptunghi:
Dreptunghiurile sunt identificate prin valori continue „0” care formează forme în cadrul matricei.
Codul utilizează urmărirea bazată pe stive pentru a identifica lățimea și înălțimea acestor dreptunghiuri.

Analiza complexității:
Complexitatea timpului:
Codul implică bucle imbricate care iterează prin matrice, rezultând o complexitate de timp de O (rânduri * coloane) pentru a procesa întreaga matrice.

Complexitatea spațiului:
Codul utilizează o matrice booleană de dimensiunea 1005x1005, rezultând o complexitate spațială de O(1005^2).
În plus, o stivă este utilizată pentru urmărirea înălțimilor, contribuind la o complexitate a spațiului proporțională cu numărul de rânduri sau coloane.

Concluzie:
Codul folosește o abordare bazată pe stivă pentru a procesa iterativ matricea, identificând dreptunghiuri formate din valori continue „0". Numărează eficient aceste dreptunghiuri în timp ce parcurge rândurile și coloanele matricei, oferind ca rezultat numărul dreptunghiurilor identificate.

Knumere

Înțelegerea problemei:
Sarcina este să găsești diferența maximă minimă de subgrupuri consecutive de dimensiune (arraySize - windowSize) într-o matrice dată.

Structura codului:
Gestionarea intrărilor:
Citește dimensiunea matricei (arraySize) și dimensiunea ferestrei (windowSize).
Inițializează un deque (indexDeque) pentru a stoca indici de diferențe.
Citește primul element (currentElement).

Bucla principală:
Iterează de la al doilea element (currentIndex = 2) până la sfârșitul matricei.
Citește următorul element (nextElement).
Calculează diferența dintre elementele curente și următoarele și o stochează în matricea de diferențe.
Actualizează deque (indexDeque) în funcție de ordinea descrescătoare a diferențelor.
Elimină indici din afara ferestrei curente din partea din față a dequei.
Actualizează minMaxDifference când se ajunge la fereastra curentă.
Actualizează elementul curent pentru următoarea iterație.

Ieșire:
Afișează rezultatul final, care este diferența maximă minimă.

Observatii:
Deque pentru stocarea indexului:

O deque (indexDeque) este utilizată pentru a menține eficient indicii diferențelor în ordine descrescătoare. Deque-ul ajută la identificarea rapidă a diferenței maxime în fereastra curentă.

Procesare bazată pe fereastră:
Codul procesează matricea într-o abordare bazată pe ferestre, actualizând deque-ul și identificând diferența maximă minimă în cadrul fiecărei ferestre.

Analiza complexității:
Complexitatea timpului:
Codul implică o singură trecere prin matrice, rezultând o complexitate de timp de O(arraySize).

Complexitatea spațiului:
Codul folosește un deque (indexDeque) pentru a stoca indici, contribuind la o complexitate spațială proporțională cu dimensiunea ferestrei (windowSize).

Concluzie:
Codul identifică în mod eficient diferența maximă minimă în cadrul subbaryurilor consecutive de o dimensiune specificată prin utilizarea unui deque pentru stocarea indexului. Utilizează o abordare bazată pe ferestre, oferind un rezultat care reprezintă diferența maximă minimă pentru toate ferestrele din matrice.