

Problema 2 summax

100 puncte

Autor Szabo Zoltan – Liceul Tehnologic „Petru Maior” Reghin

1. Soluția brută – 10 puncte

complexitate $O(2^n)$

Cu un algoritm backtracking generăm toate soluțiile și calculăm valoarea maximă a unui drum. Apoi lansăm un nou backtracking pentru a parcurge drumurile în ordine lexicografică, și tipărim în fișier soluțiile cu numerele de ordine cerute.

2. Soluție backtrack îmbunătățită – 30-45 de puncte

complexitate $O(2^n)$

Valoarea maximă a drumului se calculează cu programare dinamică cu ajutorul unei matrice.

$$\begin{aligned} b[n][j] &= a[n][j] \quad \text{pentru orice } 1 \leq j \leq n \\ b[i][j] &= a[i][j] + \max(b[i+1][j], b[i+1][j+1]), \text{ pentru orice } 1 \leq j < n \end{aligned}$$

Valoarea drumului maxim este $b[1][1]$, vom genera cu backtracking numărul drumurilor, respectiv drumurile cu numerele de ordine cerute.

3. Programare dinamică – 65 de puncte

complexitate $O(n^2)$,

memorie pentru trei matrice

Numărul drumurilor de valoare maximă se calculează într-un tabel, cunoscând matricile a și b.

$$\begin{aligned} c[n][j] &= 1 \quad \text{pentru orice } 1 \leq j \leq n \\ c[i][j] &= c[i+1][j], & \text{dacă } b[i+1][j] > b[i+1][j+1], & \text{ pentru orice } 1 \leq j < n \\ c[i][j] &= c[i+1][j+1], & \text{dacă } b[i+1][j] < b[i+1][j+1], & \text{ pentru orice } 1 \leq j < n \\ c[i][j] &= c[i+1][j] + c[i+1][j+1], & \text{dacă } b[i+1][j] = b[i+1][j+1], & \text{ pentru orice } 1 \leq j < n \end{aligned}$$

Dacă valoarea depășește 2000000000, atunci valoarea va fi 2000000001.

$c[1][1]$ va conține numărul drumurilor de valoare maximă

Cu ajutorul acestei matrice putem recalcula drumul cu un număr de ordine dat, știind câte drumuri de valoare maximă se continuă din $a[i][j]$ către $a[i+1][j]$ și câte drumuri se continuă către $a[i+1][j+1]$.

Memoria disponibilă de 16 Mo permite valoarea maximă a lui $n=1000 - 1100$

4. Programare dinamică – 80 de puncte

complexitate $O(n^2)$,

memorie pentru două matrice

Observăm că putem economisi o matrice, pentru că valorile lui a se poate deduce din valorile lui b.

Memoria disponibilă de 16 Mo permite valoarea maximă a lui $n=1400 - 1400$

5. Programare dinamică – 100 de puncte

complexitate $O(n^2)$,

memorie pentru o matrice

Atunci când declară două matrice pentru două triunghiuri, practic o jumătate de matrice este nefolosită.

Pentru a folosi memoria disponibilă în mod eficient, putem folosi două trucuri:

- a. linearizarea matricei – memorarea elementelor într-un șir
- b. cele două jumătăți de matrice se vor combina într-o singură matrice de dimensiuni $n*(n+1)$

Memoria disponibilă de 16 Mo permite valoarea maximă a lui $n=2000$