Descrierea soluției – pdl (partiție pe două linii)

Propunător : prof. Cheşcă Ciprian
Liceul Tehnologic "Grigore C. Moisil", Buzău

Considerații preliminare

Din modul de definire a partiției pe două linii putem deduce că prima linie nu poate avea un număr de termeni mai mici decât a doua linie și că un termen nu poate apărea de mai mult de două ori în partiție, deoarece toți termenii partiției sunt strict descrescători (atât pe prima linie cât și pe a doua linie). Deasemenea atât prima linie cât și a doua linie trebuie să conțină cel puțin un termen.

Soluția 1 (Programare dinamică)

student Cosmin Piț-Rada
Universitatea București

(1) Se încearcă extinderea unei configurații PDL, prin adăugarea de noi elemente.
Se observă că este extrem de benefic să folosim pe rând numerele în ordine descrescătoare, N, N-1, N-2, ... 1, întrucât inegalitățile sunt satisfăcute în mod natural.
(2) Astfel, se disting 4 tipuri de operații, la un pas i:

a) nu folosesc elementul curent i:
```
******
***
```
b) adaug elementul i doar la șirul A:
```
*******
***
```
c) adaug elementul i doar la șirul B:
```
******
****
```
d) adaug elementul i la ambele șiruri:
```
*******
****
```

(3) Pentru a caracteriza o structura PDL avem nevoie să știm:

a) câte elemente are șirul A
b) câte elemente are șirul B
c) care este suma curentă totală s, a elementelor din A și B
d) pasul curent i

Avem nevoie de a) și b) pentru a ne asigura că șirul A are cel puțin aceeași lungime ca și B.
Se poate folosi mai avantajos o valoare d, reprezentând diferența dintre lungimea lui A și cea a lui B, ceea ce înseamnă că vom folosi cu un index mai puțin.

(4) Ajungem astfel la definiția:

ways(i, s, d) = numărul de structuri PDL, în care s-au folosit elemente >= i, având suma asociată s, iar A având cu d elemente mai mult decât B.

(5) Recurența este imediată pe baza punctului (2):

ways(i, s, d) = ways(i+1, s, d) + ways(i+1, s-i, d-1) + ways(i+1, s-i, d+1) + ways(i+1, s-2*i, d)
ways(N+1, 0, 0) = 1

Definiția de mai sus necesită un fix, întrucât problema nu permite ca șirul B să fie vid. Astfel se mai calculează similar:

cnt(i, s) = numărul de structuri PDL cu B vid, folosind elemente >= i, iar suma asociată s.

cnt(i, s) = cnt(i+1, s) + cnt(i+1, s-i)

cnt(N+1, 0) = 1

Rezultatul problemei va fi: (ways(1, N, 0) + ways(1, N, 1) + ways(1, N, 2) + ... + ways(1, N, N)) - cnt(1, N)

Evident toate calculele se vor face modulo numărul din enunț și vom considera absenți termenii pentru care indecșii devin negativi sau mai mari decât N.

Recurența de față se poate rezolva comfortabil în O(N^3) timp și spațiu.

Spațiul se poate reduce la O(N^2), lucrând pe aceeași matrice ways(s, d), dacă rezolvăm recurența în ordinea descrescătoare a lui s.

Similar cnt(i, s) poate fi redus la o singură dimensiune, cnt(s).

(6) O primă observație arată că d nu poate fi foarte mare. De exemplu dacă prin absurd am avea d > sqrt(2*N) = x, atunci șirul A ar avea cel puțin x elemente distincte, a căror sumă este cel puțin 1 + 2 + 3 + ... x > x*x/2 = N, ceea ce ar conduce la o sumă totală mai mare decât N.

Așadar conchidem că d <= sqrt(2*N), deci spațiul se reduce la O(N*sqrt(N)), iar complexitatea temporală la O(N^2*sqrt(N)).

(7) O altă observație similară, arată că la un anumit pas i, nu are sens să umplem toată tabela ways(*,*) întrucât din nou d nu poate fi mai mare decât N/i.

Presupunând prin absurd că avem un d valid, d > N/i ar implica existența în șirul A a mai mult N/i elemente distincte, toate >= i, conform celor stabilite mai sus, rezultând într-o sumă > N/i * i = N, absurd. Această observație arată că la pasul i vom calcula doar N*N/i valori, ceea ce ar implica o complexitate totală O(N^2*log(N)).

(8) Evident observația (7) este slabă întrucât pentru i = 1 sugerează că d <= N, dar de fapt am observat la (6) că o margine mai bună este sqrt(2*N).

Fixăm un pas i și un d asociat. Întrucât A are cel puțin d elemente, considerând doar cele mai mici d elemente, i, i+1, ... i+d-1, suma acestora s(i, d) = i + (i+1) + (i+d-1) = d*(2*i + d - 1)/2 nu trebuie să depășească N.

Astfel pentru un i fixat, putem determina valoarea maximă a lui d, dmax, astfel încât s(i, dmax) <= N. În aceste condiții la pasul i vom calcula doar N*dmax valori.

Se observă că în timp ce i descrește, dmax va trebui să crească, așa că se poate calcula dmax asociat pasului următor i-1, incrementând succesiv valoarea curentă dmax.

Astfel putem scăpa de o căutare binară, sau o ecuație de gradul 2, făcând implementarea mult mai comfortabilă.

Se observă că dmax reprezintă o margine mult mai bună decât (6) și (7).

Noul algoritm va fi situat undeva între O(N^2) și O(N^2*(log(N)), poate chiar O(N^2).

De exemplu pentru N=1000 se calculeză ~ 5.000.000 valori în tabela ways(*,*).

Cu atenție, putem limita și indexul s, întrucât s >= i comform recurenței, numărul de valori calculate devenind ~3.500.000. Extrem de aproape de N^2.

Solutia 2 (diferență de număr de partiții)

prof. Cheşcă Ciprian
Liceul Tehnologic "Grigore C. Moisil", Buzău

Vom folosi următorul rezultat: *Numărul partițiilor pe două linii este egal cu diferența dintre numărul parțiilor crescătoare* și *numărul partițiilor strict crescătoare*.
De exemplu pentru N = 5 avem:

- 4 partiții pe două linii

$$\begin{matrix}4\\1\end{matrix}\ ;\ \begin{matrix}3\\2\end{matrix}\ ;\ \begin{matrix}3\ 1\\1\end{matrix}\ ;\ \begin{matrix}2\ 1\\2\end{matrix}$$

- 7 partiții crescătoare:
  1 + 1 + 1 + 1 + 1
  1 + 1 + 1 + 2
  1 + 1 + 3
  1 + 2 + 2
  1 + 4
  2 + 3
  5
- 3 partiții strict crescătoare
  5
  1 + 4
  2 + 3

Soluția presupune determinarea numărului de partiții crescătoare din care scădem numărul de partiții strict crescătoare.

Demonstrația acestui interesant rezultat matematic a fost dată de d-nul prof. Ionel Vasile Piț-Rada astfel:
Fie P(n) numărul partițlor descrescătoare ale lui n și Q(n) numărul partițiilor strict descrescătoare ale lui n.
Avem:
(1) P(n)=p(n,1)+ p(n,2)+...+p(n,n), unde
(2) p(n,m)= p(n-m,m-1)+ p(n-m,m-2)+...+ p(n-m,1)
(3) p(n-1,m-1)= p(n-m,m-2)+...+ p(n-m,1)
(4) din (2) și (3) rezultă p(n,m)=p(n-m,m-1)+p(n-1,m-1), cu p(n,1)=1 și p(n,n)=1 și p(n,m)=0 dacă n<m
(5) orice șir a[1]>a[2]>a[3]>...>a[m]>=1 cu a[1]+a[2]+a[3]+...+a[m]=n este echivalent cu
a[1]-(m-1)>=a[2]-(m-2)>=a[3]-(m-3)>=...>=a[m-1]-1>=a[m]-0 cu suma termenilor egală cu n-m*(m-1)/2
deci numărul partițiilor strict descrescătoare ale lui n este egal cu numărul partițiilor descrescătoare ale lui n-m*(m-1)/2, altfel spus
Q(n)=p(n,1)+p(n-1,2)+p(n-3,3)+p(n-6,4)+...+p(n-m*(m-1)/2,m)+..., până m*(m-1)/2 va depăși n
(6) rezultatul final va fi P(n)- Q(n)

Soluția 3 (Memoizare)

prof. Piț-Rada Ionel Vasile
Colegiul Traian, Drobeta – Turnu Severin

Orice partiție pe două linii va avea forma:

x_1	x_2	…	x_p	x_{p+1}	x_{p+2}	…	x_{p+q}
y_1	y_2	…	y_p				

unde

(1) $x_i>x_j$ și $y_i>y_j$ și $x_i\geq y_i$ și $\sum_{i=1}^{p+q} x_i + \sum_{i=1}^{p} y_i = S$

Forma aceasta ne permite o abordare recursivă.

Notăm cu g(S,a,b) numărul tuturor partițiilor pe două linii care respectă condițiile (1), adică încep în prima coloană cu perechea (a,b) cu a≥b și au suma elementelor egală cu S.

Notăm cu f(S,a) numărul partițiilor pe o singură linie care încep cu valori mai mici strict decât a și au suma elementelor egală cu S.

Observăm că avem $g(S,x_1,y_1)=f(S-x_1-y_1,x_1)+\sum g(S-x_1-y_1,x_2,y_2)$, unde $x_1\geq y_1$ și $x_1+y_1+x_2+y_2\leq S$ și $x_1>x_2$ și $y_1>y_2$ și $x_2\geq y_2$.

Relația de recurență provine din observația că forma $g(S,x_1,y_1)$ poate conține o singură coloană urmată de o partiție pe o singura linie, respectiv cel puțin două coloane.

Misiunea noastră este de a calcula $\sum_{S\geq x_1\geq y_1\geq 1} g(S,x_1,y_1)$.

Observăm că avem

(2) f(S,a) = f(S-(a-1),a-1)+f(S-(a-2),a-2)+…+f(S-1,1), deoarece orice partiție pe o singură linie poate începe cu a-1, a-2,…,1

(3) f(S,a-1) = f(S-(a-2),a-2)+…+f(S-1,1)

din (2) și (3) rezultă

(4) f(S,a)=f(S-(a-1),a-1)+f(S,a-1)

Cazuri particulare:

(a) f(S,0)=1

(b) f(S,a)=0 dacă S<0 sau S>a(a-1)/2

Astfel vom precalcula valorile f(S,a), iar valorile g(S,a,b) le vom calcula recursiv folosind tehnica de memoizare.

Observație

Valoarea mică a lui N permite obținerea de punctaj maxim și prin precalcularea rezultatelor cu ajutorul altor soluții.