Problema popcorn – soluţie

Problema se traduce în modul următor: *Se dau* `N` *intervale (închise)* `[l[i], r[i]]` *cu ponderi* `w[i]` *şi* `M` *puncte. Să se aleagă cele* `M` *puncte astfel încât suma ponderilor intervalelor care intersectează măcar unul dintre cele* `M` *puncte este maximă.*

Vom sorta în prealabil intervalele după capătul lor dreapta, din motive ce se vor vedea în continuare.

Să presupunem că avem o soluţie parţială, în care ultimul punct (cel mai din dreapta) este pus pe poziţia `X`. Observaţia cheie este că, pentru a nu număra de mai multe ori, trebuie să considerăm pentru celelalte puncte doar intervalele care au capătul dreapta strict mai mic decât `X`.

Avem în minte următoarea dinamică: **`DP[i][j]` = soluţia subproblemei în care considerăm `i` puncte şi intervalele cu capăt-dreapta mai mic sau egal cu `j`.**

Relaţia de recurenţă este **`DP[i][j] = min`$_x$`{DP[i - 1][x - 1] + C(x, j)}`**, unde `C(x, j)` este suma ponderilor intervalelor cu `l[i] ≤ x ≤ r[i] ≤ j`. Complexitatea acestei soluţii este $O(N^2 * M)$, dar are potenţial pentru optimizare.

Cum optimizăm? Evident, dinamica va fi calculată variind capătul j "baleiând" de la stânga la dreapta. Vom ţine o structură de date în calitate de statut al liniei de baleiere, ce ne va determină să aflăm valorile **`Val[x] = DP[i - 1][x - 1] + C(x, j)`**. La trecerea la pasul **`j`**, vom adăuga valoarea **`DP[i - 1][j - 1]`** pe poziţia `j` şi, pentru fiecare interval **`[l[k], r[k] = j]`**, vom adăuga `w[k]` pe intervalul de poziţii `[l[k]...r[k]]`. Structura de date va trebui să poată răspundă în orice moment la întrebări de tipul: *care este valoarea maximă pe structură?*

Pentru a implementa structura, putem folosi un arbore de intervale. Acest lucru conduce la soluţia în complexitate `O(N * M * log(N))`, ce va trebui să obţină 60 de puncte. Cu toate acestea, există o implementare care rezolvă problema în complexitate mai bună (şi e mult mai rapidă în practică), pe care o vom descrie în continuare:

Se poate demonstra uşor că, dacă într-un pas în procedeul de baleiere avem `Val[x] ≤ Val[y]` cu `x < y`, `x` nu va mai fi niciodată soluţie în viitor, deci putem să îl eliminăm. Astfel, putem ţine (abstract) o **listă descrescătoare de valori**. Răspunsul se va afla mereu pe prima poziţie.

Orice incrementare pe această listă se face pe un sufix al acesteia, ceea ce va determina zero, una sau mai multe eliminări. Observăm că ne este mai uşor să ţinem această structură pe diferenţe consecutive (astfel, incrementările pe intervale se transformă în incrementări pe poziţie). Dacă la un moment dat, o diferenţă este pozitivă, va "acumula" diferenţa anterioară şi se va contopi cu aceasta. Acest lucru se poate implementa folosind păduri de mulţimi disjuncte, iar complexitatea este, teoretic

`O(N * M * log*(N))`, care este practic nediferenţiabilă de `O(N * M)`. Soluţia aceasta ar trebui să obţină 75 de puncte.

Până acum nu am discutat deloc problema unui `M` mare. Pentru a o putea ataca, va trebui să observăm că funcţia **`C(x, j)`** este Monge-concavă, ceea ce ne permite să optimizăm complexităţile de mai sus în `O(N * log(N) * log(MAXZ))`, respectiv `O(N * log*(N) * log(MAXZ))`, unde `MAXZ` este o constantă (mare), în modul următor:

Fie o valoare `Z`. Rezolvăm problema redusă, în care nu mai avem restricţie pe numărul de puncte, însă fiecare punct are o pierdere asociată `Z` (cu alte cuvinte, pentru fiecare punct adăugat vom pierde scorul `Z`). Este clar că, cu cât această valoare `Z` este mai mare, cu atât numărul de puncte din soluţie va fi mai mic (acest lucru rezultă din Monge-concavitatea funcţiei de tranziţie).

Cu alte cuvinte, putem **căuta binar** valoarea `Z` pentru care soluţia problemei reduse are `M` puncte. Implementarea soluţiei problemei reduse este foarte asemănătoare cu descrierile de mai sus. Este foarte uşor de văzut că, dacă problema redusă găseşte o soluţie de cardinal `M′` şi valoare `answer(M′)`, atunci răspunsul pentru `M′` puncte este exact `answer(M′) + M′ * Z` (cu alte cuvinte, pentru acel `M′`, problema redusă "alege optim" şi pentru problema noastră).

În funcţie de structura de date folosită ca statut pentru linia de baleiere, soluţiile vor obţine 75, respectiv 100 de puncte.