Soluții pentru problema *Echipe*

Lotul Național de Informatică, Craiova 2025

Soluție pentru când toți oamenii au exact un atribut

În cadrul acestui subtask, este ușor să determinăm numărul maxim de echipe care se pot forma (fie folosind o formulă directă, fie folosind căutare binară). Acum, putem să împărțim oamenii în patru grupe: C, I, M și IM (din mulțimea IM fac parte oamenii care sunt fie I, fie M și au fost selectați ca al patrulea membru din echipă). Știm, deci, pentru fiecare tip de oameni prefixul (dacă considerăm șirurile sortate crescător după riscuri) de oameni care fac parte dintr-o echipă.

Să notăm cu K numărul de echipe. Acum, trebuie să vedem cum putem rezolva update-urile. Dacă avem un update care schimbă un om care este de tipul C, schimbarea este simplă: doar dăm update la riscul omului și calculăm suma celor mai mici K valori de tipul C (acest lucru se poate face fie cu două seturi, fie cu un arbore de intervale sau un arbore indexat binar).

Rămâne de văzut cum rezolvăm un update care ne schimbă un om de tipul I sau M. Fie L_I lungimea prefixului de oameni de tipul I care fac parte din soluție și L_M lungimea prefixului de oameni de tipul M care fac parte din soluție. Este ușor de observat că aceste valori se schimbă cu cel mult unu la fiecare update. Putem să încercăm toate aceste schimbări care încă respectă condițiile următoare:

- $L_I + L_M = 3 \cdot K$
- $L_I \geq K$
- $L_M \geq K$

Prin urmare, putem să rezolvăm un query în $O(\log N)$, fie cu două `std::set`-uri sau cu orice structură arborescentă (de exemplu, un arbore de intervale). Complexitatea timp a acestui subtask este, deci, $O(Q \cdot \log N)$.

Soluție pentru când $1 \leq N, Q \leq 150$

Vom căuta binar numărul maxim de echipe pe care putem să le formăm. Acum, trebuie să verificăm dacă putem să formăm K echipe. Să considerăm următoarea rețea de flux cu costuri pe muchii:

- Un nod sursă (notat cu S) și un nod destinație (notat cu T).
- N noduri, al i-lea dintre ele reprezentând omul i.
- Alte 4 noduri, reprezentând rolul oamenilor în echipă: I, C, M sau IM (tipul omului care se repetă în echipă).

Când încercăm să vedem dacă putem forma K echipe, vom adăuga următoarele muchii în rețea:

- Muchie de cost $risk_i$ și capacitate 1 de la S la al i-lea om.
- Muchie de cost 0 și capacitate 1 de la omul i la toate rolurile pe care acesta le poate juca într-o echipă.
- Muchie de cost 0 și capacitate K de la cele patru noduri de rol la T.

Dacă fluxul maxim din acest graf este egal cu $4 \cdot K$, atunci putem să formăm K echipe. De asemenea, datorită costurilor de pe muchii, avem și suma riscurilor oamenilor prezenți în soluția minimă.

Soluție pentru când $Q = 0$ și $risk_i = 1$

În acest subtask, vrem doar să găsim numărul maxim de echipe pe care le putem forma. Vom pleca de la subtask-ul anterior și vom vedea cum putem îmbunătăți soluția.

Pentru subtask-ul anterior construiam un graf cu $O(N)$ noduri și $O(N)$ muchii. Vrem să folosim în continuare o rețea de flux ca să găsim răspunsul la problemă, dar vrem ca aceasta să aibă puține noduri și puține muchii (pentru a putea determina fluxul rapid).

Următoarea rețea de flux este echivalentă cu cea de mai sus:

- Un nod S pentru sursă, un nod T pentru destinație.
- 4 noduri pentru rolurile oamenilor în cadrul echipei (I, M, C sau IM).
- 7 noduri pentru fiecare subset de atribute pe care îl poate avea un om.

Vom lega muchii de la fiecare tip de om (unde tipul este subsetul de atribute al acestuia) la rolurile pe care acesta le poate juca într-o echipă (cu costul $risk_i$ și capacitate 1). Vom trage muchii de la sursă la cele 7 noduri de tip subset (unde capacitatea unei muchii este numărul de oameni din input cu acel subset de atribute). De la cele 4 noduri de rol vom trage muchiile la fel ca la subtask-ul anterior.

Din păcate, avem în continuare $O(N)$ muchii, dar am redus graful la $O(N)$ noduri. Însă, cum $risk_i = 1$ pentru toate nodurile, putem să compresăm muchiile și obținem un graf cu $O(1)$ noduri și $O(1)$ muchii. Avem 7 noduri în partea stângă și 4 noduri în partea dreaptă. De la fiecare nod din stânga vom trage muchii către nodurile din dreapta cu care este compatibil. Știm de câte ori apare fiecare tip de muchie.

Acum, trebuie să rezolvăm o problemă de cuplaj pe acest graf. Aceasta poate fi rezolvată folosind Lema Mariajelor. Mai exact, răspunsul este valoarea minimă a expresiei $\frac{neighborhood_{subset}}{size_{subset}}$, unde $neighborhood_{subset}$ este vecinătatea lui *subset* în partea opusă a grafului. Putem să iterăm prin toate cele 16 subseturi ale părții drepte și să calculăm valoarea minimă.

Prin urmare, după construcția grafului putem să calculăm cuplajul în acest graf în $O(1)$. Soluția are, deci, complexitate $O(N)$.

Soluție pentru când $Q = 0$

Vom pleca de la soluția de la subtask-ul anterior și vom determina numărul K de echipe pe care le putem forma. Trebuie acum să determinăm care este suma riscurilor minimă pentru a forma cele K echipe.

Pentru aceasta, vom folosi următorul greedy: considerăm că toți oamenii fac parte dintr-o echipă și vom încerca să îi scoatem pe rând. Sortăm oamenii în ordine descrescătoare a riscurilor și încercăm să vedem dacă putem să îl scoatem pe cel cu riscul cel mai mare. Actualizăm $neighborhood_{subset}$ și recalculăm soluția. Dacă putem în continuare să formăm K echipe, vom scoate omul din soluție. Altfel, trebuie să îl păstrăm, așa că îl adăugăm din nou în graf (actualizând din nou $neighborhood_{subset}$).

Observăm că, de vreme ce putem să calculăm numărul de echipe în $O(1)$ (odată calculat *neighborhood*), soluția de mai sus va avea complexitate $O(N \cdot \log Q)$.

Soluție completă

Vom folosi soluția anterioară pentru a determina oamenii care fac parte dintr-o echipă. Trebuie doar să rezolvăm update-urile acum.

Pentru început, vom grupa oamenii în 32 de grupe. Grupa $(subset, 0/1)$ reprezintă mulțimea oamenilor care au ca atribute *subset* și fac parte (sau nu) dintr-o echipă. Să considerăm aceste mulțimi sortate în ordine crescătoare a riscurilor.

Observația care rezolvă problema este următoarea: în momentul în care schimbăm riscul unui om, mulțimea oamenilor care fac parte din soluție se schimbă cu cel mult un om. Putem să observăm acest lucru în mod intuitiv din grupele $(subset, 0/1)$. Este clar că în cazul în care omul va rămâne în aceeași grupă, soluția nu se schimbă. În cazul în care omul iese dintr-o grupă de tipul 1 (adică făcea parte din soluție), trebuie înlocuit cu cel mai puțin riscant om dintr-un subset care este compatibil cu el. Cazul în care acesta intră în soluție este similar.

Prin urmare, putem să iterăm prin toate cele 16 subseturi și să încercăm să schimbăm omul al cărui risc se schimbă cu cel mai bun din acea mulțime. Calculăm din nou cuplajul, iar dacă acesta este tot K, ținem minte valoarea minimă.

Prin urmare, soluția are complexitate $O(N \cdot \log N + Q \cdot \log N)$, unde update-urile au constantă de 32.

Notă: Atât observația de mai sus, cât și corectitudinea greedy-ului pentru a determina configurația cu suma riscurilor minimă se pot demonstra atât elementar (demonstrația este destul de simetrică — cazurile în care scoatem oameni sunt la fel cu cele în care adăugăm oameni), însă este nevoie de atenție la toate cazurile, cât și dacă avem cunoștințe despre matroizi.