Descrierea soluțiilor

1 Problema Lungcircuit

Autor: Cătălin Frâncu, instructor, Nerdvana România

1.1 Subproblema 1: $n, q \leq 10\,000$

Putem simula naiv fiecare mutare, rotind circuitul corespunzător în $\mathcal{O}(n)$. Complexitatea totală este $\mathcal{O}(qn)$.

1.2 Subproblema 2: $k \leq 10$

Putem reprezenta fiecare circuit ca pe un buffer circular: un vector v cu n poziții și un pointer la primul element p. Atunci conținutul vectorului este $v[p], \ldots, v[n-1], v[0], \ldots, v[p-1]$.

Cu această reprezentare, putem roti un circuit în $\mathcal{O}(1)$: simulăm că bilele avansează cu o poziție scăzându-l pe p cu o poziție (modulo n). Este totuși nevoie de k atribuiri pentru a actualiza și celălalt circuit.

Complexitatea totală este $\mathcal{O}(n + qk)$.

1.3 Subproblema 3: Există doar $r \leq 10$ mutări R

Algoritmul anterior este ineficient dacă există multe mutări identice consecutive. El face degeaba k atribuiri pentru a actualiza circuitul opus, pentru că imediat va modifica din nou aceleași k poziții.

De aceea, modificăm algoritmul ca să actualizeze circuitul opus doar dacă mutarea următoare diferă de mutarea curentă (sau la final). Complexitatea totală este $\mathcal{O}(n + q + rk)$.

1.4 Subproblema 4: $n, q \leq 100\,000$

Introducem doi termeni pentru a ușura descrierea. Numim pozițiile comune *noduri*. Primul nod este la poziția 0, iar ultimul nod este la poziția $(k-1)d$. Numim *lanțuri exterioare* pozițiile $(k-1)d+1 \ldots n-1$ pe fiecare circuit.

Pentru o soluție în $\mathcal{O}((q+n)\log k)$, să schimbăm paradigma. Să urmărim traseul fiecărei bile și să aflăm poziția ei finală după cele q mutări. Vom accelera simularea procesînd rapid porțiuni de circuit. Pornind de la mutarea 0, fiecare bilă va trece prin următoarele etape (sau o submulțime a lor):

1. Avansează pînă la următorul nod.
2. Avansează pînă la ultimul nod.
3. Avansează pînă la primul nod (parcurgînd lanțul exterior).
4. Găsește ultima mutare cînd bila mai revine în primul nod.
5. Avansează un număr întreg de noduri (cel mult $k-1$).
6. Avansează pozițiile rămase pînă la epuizarea operațiilor.

Pentru (1), (3) și (6), știind că bila se află pe circuitul c la mutarea t și că dorim să avanseze p pași, trebuie să putem răspunde la întrebarea „ce indice are a p-a mutare de tip c, începînd cu mutarea t”? Putem răspunde la aceste întrebări cu o precalculare ușoară, colectînd în doi vectori pozițiile literelor **R** și **A** în șirul de mutări.

Pentru (2) și (5), să remarcăm că bila pornește dintr-un nod, deci prima mișcare dictează pe ce drum pornește bila. Pentru a avansa un interval în $\mathcal{O}(1)$, dacă prima mișcare este c la momentul t, trebuie să găsim rapid a d-a apariție a lui c, începînd cu mutarea t, în șirul de mutări. Și această informație poate fi precalculată.

Noi dorim să avansăm eficient $\mathcal{O}(k)$ intervale, deci peste informația de mai sus construim o tabelă rară (*sparse table*). Astfel putem avansa pînă la ultimul nod în $\mathcal{O}(\log k)$.

Pentru (4), dorim să răspundem la întrebări de forma: „dacă o bilă se află în nodul 0 la momentul t, cînd va reveni ea în nodul 0?”. Putem refolosi exact codul anterior. Dacă răspunsul există și este t', atunci putem completa de la sfîrșitul spre începutul vectorului de mutări momentul *ultimei* reveniri în nodul 0: el va fi același pentru t și pentru t'.

1.5 Subproblema 5: $n, q \leq 1\,000\,000$

O soluție în $\mathcal{O}(n+q)$, semnificativ mai simplu de codat, pornește de la următoarea observație. Bilele din noduri (în exemplul dat, 0, 3, 6, 9 și 12) au o traiectorie similară. Dacă prima mutare este **R** și bila 0 intră pe circuitul roșu, la fel vor face și celelalte. Cînd 0 revine într-un nod, la fel și celelalte. Similar sînt unite și bilele 1, 4, 7, 10 și 13.

De aceea, putem să grupăm aceste bile și să le mutăm pe toate cu efort mic. Concret, reprezentăm un circuit ca:

- d liste, prima de k bile, iar restul de cîte $k-1$ bile. Numim această structură *rețea*, căci elementele listelor sînt intercalate. Prima listă conține nodurile și este comună celor două circuite.
- Lanțul exterior (inițial 16-31, respectiv 42-57).

Ce se întîmplă la o mutare **R** (cazul **A** este simetric)?

- Lanțul exterior capătă o bilă la început, și anume ultimul nod (15).
- Lanțul exterior pierde o bilă de la final (31), care ajunge la începutul ultimei liste din rețea, care devine 31, 2, 5, 8, 11, 14.
- Permutăm circular listele rețelei. Ultima listă devine prima. Astfel, bila 31, care acum stă în nodul 0, devine în mod corect prima bilă din prima listă.
- Notificăm circuitul albastru că prima sa listă nu mai este $0, 3, \ldots$, ci este $31, 2, \ldots$

Cu această reprezentare putem simula toți pașii de mai sus, și deci toate mutările, în $\mathcal{O}(1)$ fiecare. Putem reprezenta atît listele din rețea, cît și lista de liste (rețeaua însăși), ca pe buffere circulare, ca să le putem permuta circular în $\mathcal{O}(1)$.

Unele implementări se simplifică considerabil dacă stocăm ultimul nod separat de rețea. Atunci toate listele din rețea au lungime egală, $k-1$.

Implementarea cu `deque`-uri este chiar mai scurtă, dar este foarte ineficientă cînd d este mare, deoarece creează o mulțime de `deque`-uri mici.