

Trip

Autor: stud. Floare Doru

Contribuitori: stud. Măierean Mircea, stud. Alexuțan Cristian, Universitatea Babeș-Bolyai Cluj-Napoca

Soluție

Înlocuind caracterele din input cu valorile $-1, 0, 1$, putem reformula query-urile sub forma: există o secvență continuă care începe în intervalul $[l_1, r_1]$ și se termină în intervalul $[l_2, r_2]$, cu suma k ? Calculând sume parțiale, fie $s[i]$ suma primelor i elemente din șir, query-ul se transformă în: "există un $i, l_1 - 1 \leq i \leq r_1 - 1$ și un $j, l_2 \leq j \leq r_2$, astfel încât $s[j] - s[i] = k$ ", iar update-ul devine un update pe interval, deoarece valoarea de la o poziție p influențează toate valorile sumelor parțiale din intervalul $[p, n]$.

Pentru rezolvarea query-urilor avem nevoie de o observație importantă: deoarece elementele din șir fac parte din mulțimea $\{-1, 0, 1\}$, valorile sumelor parțiale sunt "continue" în mulțimea numerelor întregi. Adică, dacă într-un interval l avem valoarea maximă M și valoarea minimă m , atunci acel interval conține toate numerele întregi din $[m, M]$. Acum putem determina, pentru cele două intervale din fiecare query valorile minime și maxime, fie acelea $[m_1, M_1]$ pentru intervalul $[l_1 - 1, r_1 - 1]$ și $[m_2, M_2]$ pentru intervalul $[l_2, r_2]$. Folosind acest raționament, aflăm că valorile posibile ale diferenței dintre o sumă parțială din intervalul 2 și una din intervalul 1 se află în intervalul $[m_2 - M_1, M_2 - m_1]$. Rămâne doar să verificăm dacă k face parte din acest interval, iar dacă face, răspunsul este 1, altfel 0.

Maximele, minimele și update-urile pe interval pot fi rezolvate folosind un arbore de intervale cu lazy update, astfel complexitatea fiind $O(n \log(n))$.