Padure

Autor: stud. Lăuran David-Andrei, Universitatea Babeș-Bolyai, Cluj-Napoca

Problema cere aflarea maximului sumei

$$S = \sum_{\substack{2 \le i \le n \\ q_{i-1} > q_i}} q_i,$$

unde q este o permutare a lui p. Cu alte cuvinte, S este suma numerelor de pe pozițiile i ($i > 1$) care au proprietatea că $q_{i-1} > q_i$, iar nouă ni se cere să aflăm maximul pe care S îl atinge în cele $n!$ variante de a-l alege pe q.

q poate fi construit folosind un algoritm greedy: vom face N „pași", pasul numărul i reprezentând alegerea lui q_i. La fiecare pas alegem cel mai mare număr disponibil care este strict mai mic decât numărul anterior. Dacă nu există niciun număr strict mai mic decât numărul anterior sau dacă suntem la primul pas, alegem cel mai mare număr dintre toate numerele disponibile.

Soluție $O(N * max(p_1, p_2, ..., p_N))$

Algoritmul greedy descris mai sus poate fi implementat astfel:

Fie $P = max(p_1, p_2, ..., p_N)$ și $f_i =$ numărul de apariții ale lui i în p, $i = 1, 2, ..., P$. Prima dată, calculăm șirul f printr-o parcurgere a lui p. Pe urmă, parcurgem descrescător valorile de la P la 1 (P, $P-1$, $P-2$, ..., 1) și, pentru fiecare valoare, în cazul în care frecvența sa este mai mare decât 0, punem valoarea la finalul lui q și îi scădem cu 1 frecvența. Repetăm această parcurgere până când frecvențele tuturor valorilor sunt 0. La final, calculăm S parcurgând șirul construit și adăugând la S acele numere care sunt precedate de un număr strict mai mare. Numărul de parcurgeri este egal cu frecvența maximă ($max(f_1, f_2, ..., f_P)$).

Această soluție acoperă testele din primul subtask ($N \le 10^3$ și $p_i \le 10^4, \forall i \in [1, n]$).

Soluție $O(max(p_1, p_2, ..., p_N))$

În soluția anterioară, la fiecare parcurgere a valorilor de la P la 1 adăugăm la finalul lui q un șir strict descrescător. Este posibil să facem o singură parcurgere a valorilor de la P la 1, în cadrul căreia vom „construi" separat acele șiruri descrescătoare.

Notăm numărul curent de șiruri cu M. Inițial, $M = 0$. Parcurgem valorile de la P la 1 și, pentru fiecare valoare v, procedăm în felul următor:

1. dacă $f_v \le M$, atunci punem v la finalul a f_v șiruri din cele M. Știm că în toate cele M șiruri avem doar numere mai mari decât v, așa că v poate fi pus la finalul oricăruia dintre șiruri (cel mult o dată în fiecare șir). Astfel, v se va afla tot timpul după un număr mai mare decât el, deci va fi adăugat la sumă.

2. dacă $f_v > M$, atunci punem v la finalul tuturor celor M șiruri existente, apoi creăm $f_v - M$ șiruri noi care îl conțin doar pe v. Astfel, îl vom adăuga pe v la sumă de M ori.

Observăm că în cazul 1., S crește cu $v \cdot f_v$ și M rămâne neschimbat, iar în cazul 2., S crește cu $v \cdot M$ și M devine f_v. Astfel, nu este nevoie să construim șirurile, fiind suficient să ținem minte valorile lui M și S și să le actualizăm la fiecare dintre cei P pași.

Această soluție acoperă testele din primele două subtaskuri (cel acoperit de soluția anterioară și cel care garantează $p_i \le 10^7, \forall i \in [1, n]$).

Soluție $O(N \cdot log\, N)$

În soluțiile anterioare, parcurgem P valori, dintre care cel mult N valori apar în șirul p, celelalte având frecvența 0 și fiind irelevante. Cum N este, în general, mai mic decât P, următoarea soluție este superioară:

Sortăm p descrescător, apoi îl parcurgem. În timpul parcurgerii calculăm frecvențele și le procesăm conform cazurilor 1. și 2. din soluția anterioară. Pentru a calcula frecvențele în timpul parcurgerii, procedăm în felul următor pentru fiecare dintre cele N numere: dacă numărul curent este egal cu cel anterior, creștem frecvența curentă. Altfel, frecvența curentă reprezintă frecvența reală a valorii de pe poziția anterioară, așa că o vom procesa conform cazurilor 1. și 2., apoi vom reseta frecvența curentă la 1. În acest mod, parcurgem descrescător toate valorile care apar în p.

Această soluție obține punctaj maxim.

Soluție $O(N)$, pentru cazul în care $p_1, p_2, ..., p_N$ sunt distincte

Cea mai mare valoare din p nu va fi niciodată inclusă în sumă, deoarece în p nu există o valoare mai mare decât ea. Astfel, $S \leq p_1 + p_2 + ... + p_N - max(p_1, p_2, ..., p_N)$ pentru orice permutare a lui p. Notăm cu T valoarea din membrul drept.

Fie r = șirul p sortat descrescător. Folosindu-l pe r, obținem $S = r_2 + r_3 + ... + r_N$, deoarece $r_{i-1} > r_i, \forall i \in [2, n]$. Dar $r_1 = max(p_1, p_2, ..., p_N)$, deci $S = r_1 + r_2 + ... + r_N - max(p_1, p_2, ..., p_N) = p_1 + p_2 + ... + p_N - max(p_1, p_2, ..., p_N) = T$. Așadar, există o permutare a lui p pentru care S atinge valoarea T.

Am arătat că $S \leq T$ pentru orice permutare și că există o permutare pentru care $S = T$. Deducem că valoarea maximă pe care S o poate atinge este T.

Această soluție acoperă al treilea subtask ($p_1, p_2, ..., p_N$ sunt distincte).