

Concursul Județean de Informatică „Future for Future”
Ediția a II-a, 2025
Clasele VII-VIII
Descrierea soluțiilor

1. Problema Ursus

Propusă de: Ștefan Patrichi, CNMB

Având în vedere faptul că g , numărul de generații, este relativ mic față de q , putem precalcula numărul de urși din fiecare generație într-un vector, fie acesta `gen[]`, astfel încât `gen[i]` reprezintă numărul de urși din generația i .

Pentru cerința 1, este suficient să afișăm elementul corespunzător din vectorul `gen[]`, complexitate timp $\mathcal{O}(g + q)$.

Pentru cerința 2, ținem cont de faptul că elementele vectorului `gen[]` sunt sortate crescător și căutăm binar cea mai mare valoare din vector mai mică sau egală cu n , complexitate timp $\mathcal{O}(g + q \log g)$.

Exemplu de implementare: <https://kilonova.ro/pastes/A0UDzvy2QH08>

2. Problema Tenu

Propusă de: George-Mihai Tega, CNMB

Restul pe care curierul trebuie să i-l dea lui David este $r = 5^k - (ab \% 5^k)$, cu mențiunea că atunci când $5^k \mid ab$, restul este 0.

Pentru primul subtask calculăm direct r , pentru cel de-al doilea ne folosim de distributivitatea `%` față de \cdot pentru produsul ab , iar pentru cel de-al treilea putem stoca cifrele numerelor într-un vector, aplicând apoi algoritmul clasic de înmulțire „de mână”.

Pentru 100 de puncte este necesară observația că este suficient să luăm în considerare doar ultimele k cifre ale lui a . Într-adevăr, $a = 10^k a_1 + a_2$, unde a_1 și a_2 sunt numerele formate din primele $n - k$, respectiv ultimele k cifre ale lui a . Văzând această egalitate modulo 5^k , cum $5^k \mid 10^k$, constatăm că $a \equiv a_2 \pmod{5^k}$.

O soluție alternativă, dar mai greu de implementat, constă în folosirea unei metode similare cu cea de la subtaskul 2 pentru a calcula $a \% 5^k$ și $b \% 5^k$.

Exemplu de implementare: <https://kilonova.ro/pastes/1njbB34570JM>