



## Problema Ținut

Clasa a X-a  
Autori Stud. Alexandru-Paul Sîrbu, Stud. Andrei-Paul Bejan  
Universitatea Babeș Bolyai, Cluj-Napoca

### Cerința 1

La prima cerință rezolvarea constă în determinarea numărului de divizori, apoi aplicarea algoritmului lui Lee pentru a determina soluția problemei. În funcție de modalitatea de determinare a divizorilor se pot obține diferite punctaje, cea mai eficientă dintre ele constă în precalcularea ciurului lui Eratosthenes și apoi determinarea descompunerii numărului în factori primi. Pentru a obține toate punctele aferente cerinței se va folosi faptul că numărul de divizori pentru un număr  $N$  ce se scrie descompus în factori primi ca  $p_1^{k_1} p_2^{k_2} \dots p_n^{k_n}$  este  $(k_1 + 1) * (k_2 + 1) * \dots * (k_n + 1)$ .

### Cerința 2

Pentru a doua cerință orice se va folosi algoritmul lui Lee cu costuri. Este necesar să ținem în permanență sortate costurile, deci va trebui să ne folosim de un container specific (putem alege din containerele STL cum ar fi `priority_queue` sau `set`).

### Cerința 3

Pentru a treia cerință se va căuta cea mai mică valoare posibilă pentru care putem traversa toată matricea, în siguranță. Căutarea valorii poate fi făcută liniar, în complexitate  $O(n)$ , vom parcurge tot intervalul (de la cel mai mic la cel mai mare, prima valoare pentru care se poate face traversarea fiind rezultatul dorit). Datorită dimensiunii mari a datelor de intrare această cerință va obține doar 10 puncte. Pentru a obține toate punctele, păstrând o parte din ideea de mai sus, vom schimba modalitatea de căutare, aplicând căutarea binară, astfel complexitatea căutării reducându-se la  $O(\log n)$ .