Descriere problema showroom

Prof. Dr. Doru Anastasiu Popescu, C.N.”Radu Greceanu”, Slatina

Varianta 1

Soluţie propusă de Prof. Dr. Doru Anastasiu Popescu

Trebuie sa construim o structură de date cu denumirile modelelor pentru fiecare dealer.

Se citeşte pe rand câte o linie din fişier, se identifică cuvintele distincte si se determină dealerul căreia îi aparţin modelele (dacă există). În caz afirmativ se completează dealerul cu noile modele, altfel se adaugă o un nou dealer cu aceste modele.

În felul acesta se obţin:

p dealeri cu câte x_1, x_2, ..., x_p modele fiecare. La punctul a) trebuie să afişăm p.

Notând cu f_1, f_2, ..., f_p primii p termeni modulo 9001 din progresie, la punctul b) trebuie să afişăm valoarea expresiei:

$C_{x[1]}^{f[1]} \cdot C_{x[2]}^{f[2]} \cdot \ldots \cdot C_{x[p]}^{f[p]}$ modulo 9001.

In functie de algoritmul folosit pentru rezolvarea cerinţei a) şi de modul de calcul al numărului de combinări de la cerinţa b) se pot obţine punctaje diferite.

Varianta 2

Soluţie propusă de dr. Csaba Pătcaş

Pentru a îmbunătăţi timpul de rulare, putem înlocui fiecare model de maşină cu un număr unic deja din faza citirii datelor. Pentru o soluţie eficientă şi o implementare simplă se pot folosi tipurile set şi map din biblioteca STL.

Pentru a determina modelele care aparţin fiecărui dealer fără a parcurge liniile din fişierul de intrare de mai multe ori, putem aplica următoarea idee. Fie *m* numărul modelelor distincte, numerotate de la 1 la *m*, valori determinate în pasul precedent. Numerotăm de la *m*+1 la *m*+*n* liniile în care sunt amplasate modelele. Pentru fiecare dintre cele *m*+*n* entităţi obţinute astfel, reţinem două valori: **parent[i]**, semnificând indicele unei entităţi de care aparţine entitatea *i* (sau 0, dacă nu aparţine de nicio entitate), respectiv **size[i]**, numărul modelelor distincte ce aparţin entităţii *i*. Un model poate aparţine unei linii, iar o linie la rândul ei poate aparţine altei linii.

Când parcurgem linia *j*, reţinem valoarea **this**, care semnifică numărul entităţii de care aparţin modelele din linia actuală. La început iniţializăm **this** cu *m* + *j.* Când întâlnim un model *i* cu **parent[i] = 0**, asta înseamnă că este vorba de un model pe care nu l-am mai întâlnit. Setăm **parent[i] = this** şi incrementăm **size[this]**.

În cazul în care **parent[i]** este diferit de zero, trebuie să unim entităţile ce aparţin de *i* cu entităţile ce aparţin de **this**. Urmărim valorile din **parent** pornind din *i* şi din **this**, pentru a ajunge la entităţile cele mai importante care le aparţin (cele care au valorea din **parent** egală cu zero), fie acestea *x* şi *y*. Dacă *x* este diferit de *y*, trebuie să creem o legătură între ele prin intermediul şirului **parent**. Pentru a obţine un timp de rulare cât mai mic, comparăm **size[x]** cu **size[y]** şi modificăm valoarea **parent** corespunzător celui mai mic dintre cele două.

Subpunctul a) îl putem rezolva prin iterarea peste cele *n* linii și găsirea entității cele mai importante de care aparține prima mașină din linia actuală. Între timp reținem într-o mulțime entitățile la care am ajuns deja. Dacă este vorba de o entitate nouă, înseamnă că am găsit un dealer nou, introducem entitatea în mulțime și citim numărul modelelor pe care le are dealer-ul din șirul **size.**

Există mai multe abordări pentru rezolvarea subpunctului b). Ideile bazate pe construcția triunghiului lui Pascal depășesc timpul și / sau memoria pentru testele mari. Abordarea cu invers modular iarăși întâmpină probleme, din cauză că numărul **9001** nu este suficient de mare pentru a acoperi valorile din denumitor la formulele de combinări pentru fiecare test.

Pentru obținerea punctajului maxim se poate folosi următoare abordare. Fie C_n^k unul dintre combinări ce trebuie calculate. Vom determina numerele prime până la cea mai mare valoare n pe care o putem întâlni (practic numărul maxim de mașini, care aparțin unui dealer), folosind metoda ciurului lui Erastotene. Pe urmă în șirul **power** la poziția *i* reținem puterea la care se află numărul prim *i* în valoarea C_n^k.

Pentru a determina valorile corecte din **power**, ne folosim de formula n!/k!(n-k)!. Prima dată iterăm cu *j* de la 1 la *n,* îl descompunem pe *j* în factori primi și incrementăm valoarea **power[p]** pentru fiecare factor prim *p*, cu exponentul la care acesta apare în descompunere. Pe urmă repetăm procedeul de la 1 la *k* și de la 1 la *n-k*, cu diferența că de această dată decrementăm valorile din **power**.

La sfârșit ne rămâne se iterăm de la 1 la *n* încă o dată și pentru fiecare *i* să înmulțim soluția cu $i^{power[i]}$.