DESCRIERE SOLUŢII

PROBLEMA 1 ALARMA

Autori: prof. Violeta și Radu Vişinescu
Colegiul Naţional "I.L.Caragiale", Ploieşti

Soluţia 100 puncte
Problema se rezumă la găsirea momentului de timp care apare în prima programare cu frecvenţă maximă. Se poate folosi o matrice de frecvenţă cu 24 de linii şi 60 de coloane sau un vector de frecventă obşinut prin liniarizarea matricii.

PAS 1. Se parcurge liniar şirul momentelor de timp din fişierul de intrare şi se actualizează matricea A[h][M], pentru h şi M dintre cele N momente considerate.

PAS 2. Se găseşte elementul maxim al matricei, precum şi numărul său de apariţii.

PAS 3. Se tipăresc pe rând linia şi coloana de matrice pentru care în matricea A se atinge acel maxim.

Pasul 1 necesită O(N*24*60) operatii, paşii 2 şi 3 necesită O(24*60)~O(1) operaţii. Aşadar se poate spune că algoritmul este liniar în dimensiunea datelor de intrare, dată fiind limita pentru N .

PROBLEMA 2 DIVCOM

Autor: eleva Diana Ionița
Colegiul Naţional "Mihai Viteazul", Ploieşti

Soluţia 20 puncte
Pentru fiecare număr de la 2 la valoarea maximă care se regăseşte în şir se contorizează câte numere sunt divizibile cu el. Se afişează cea mai mai mare valoare găsită.

Soluţia 40 puncte
Obsevaţie: dacă cmmdc(a,b)=c, atunci toţi divizorii primi ai lui c îi divid pe a şi b.
Datorită faptului că nu se cere valoarea celui mai mare divizor comun, nu este necesar să determinăm numărul de elemente divizibile cu el, pentru valorile care nu sunt prime. Pornind de la această observaţie, descompunem fiecare număr din şir în factori primi, iar cu ajutorul unui vector de frecvenţă se determină pentru fiecare număr prim în descompunerea câtor elemente se regăseşte. Lungimea maximală a subşirului este maximul din vectorul de frecvenţa.

Soluţia 100 puncte
Pornind de la ideea din soluţia 1 şi observaţia precedentă se deduce că soluţia optimă va contoriza pentru fiecare număr prim mai mic decât 500 000 câte elemente din şir sunt divizibile cu el şi se va afişa valoarea maximă dintre aceste rezultate.

PROBLEMA 3 MAGAZIN

Autor: student Andrei Drăgan
Universitatea Babeş- Bolyai, Cluj Napoca

Pentru început, pentru fiecare zi în parte, se va afla preţul tortului de îngheţată astfel:

- se citeşte **T**;

- se calculează **X** = $\mathbf{2^T}$ în timp logaritmic, având grijă să se realizeze și operația modulo;
- se caută binar în șirul **A** pe ce poziție poate fi plasat **X**, astfel obținându-se prețul tortului de înghețată din ziua respectivă.

Vom continua prin a face următoarea observație – dacă suma totală de bani necesară pentru a cumpăra înghețată în intervalul de zile [**1-i**] și suma totală de bani necesară pentru a cumpăra înghețată în intervalul de zile [**1-j**] au același rest la împărțirea prin **K**, atunci intervalul de zile [**(i+1)-j**] constituie un interval de zile valid.

De ce?
Să notăm suma totală de bani necesară pentru a cumpăra înghețată în intrvalul de zile [**1-i**], S_i. Vom proceda la fel și pentru **j**, obținând astfel S_j. Conform observației de mai sus, putem vedea că $S_i = K * c_1 + r$ iar $S_j = K * c_2 + r$. Astfel, prin scădere, se obține că suma totală din interval [**(i+1)-j**] este egală cu $S_j - S_i = K * (c_2 - c_1)$, sumă ce poate fi plătită de Budi printr-un număr exact de bancnote de valoare **K**.

Astfel, pe baza acestei observații, vom reține pentru fiecare zi **i**, suma totală de bani necesară pentru a cumpăra înghețată în intervalul de zile [**1-i**]. Bineînțeles, pentru fiecare zi, aceasta sumă se va actualiza, pastrându-se mereu restul împărțirii la **K** - $suma_i = (suma_{i-1} + Y) \% K$, unde **Y** reprezintă prețul tortului de înghețată din ziua **i**.

Problema se reduce acum la două cazuri:

1. Nu am obținut încă acest rest, caz în care păstram într-un vector **fv** prima sa apariție.

2. Am obținut anterior acest rest - $i_{anterior}$ = **prima sa apariție**. Astfel, obținem o secvență de zile validă, reprezentată de intervalul [$(i_{anterior}+1)-i$]. Se compară lungimea acesteia cu maximul și se actualizează soluția.

Se va lua în considerare și ziua 0, care va reprezenta prima apariție a restului 0.

Complexitate: O(N * (log(T) + log(Q))

PROBLEMA 4 EXPEDITIE

Autori: elev Andrei Băzăvan
elev Ioan Mihai
Colegiul Național “Mihai Viteazul”, Ploiești

Fiecărei stații îi vom atribui un set de informații, distanța la care se află, prețul unui bilet, ora sosirii, ora plecării și un șir ordonat alfabetic cu inițialele obiectelor.

La citirea stațiilor vom ordona alfabetic șirurile de litere și le vom compara cu șirul inițial cu obiectele pe care Marcel dorește să le ia, iar dacă găsim un obiect ce aparține ambelor șiruri îl contorizăm, la final, dacă numărul obiectelor contorizate este mai mare sau egal cu numărul minim de obiecte pe care Marcel dorește să le ia cu el, marcăm stația ca fiind validă.
Sortăm stațiile în ordine crescătoare după distanța la care se află.

Pentru C=1:

Verificăm fiecare tren de la 1 la n ce se află la o distanță mai mică decât cea impusă dacă este valid. Reținem cea mai îndepărtată stație validă și o afișăm, în cazul în care nu s-a găsit nicio stație afișăm mesajul: EXPEDITIE ESUATA :(.

Pentru C=2:

Pentru fiecare stație de la 1 la n verificăm dacă este validă (Marcel poate lua numărul minim de K obiecte cu el), verificăm dacă nu depășește distanța limită D, iar dacă îndeplinește ambele condiții, verifcăm dacă, luând acest tren și la dus și la întors prețul ambelor bilete nu depășește bugetul S, în caz contrar căutăm o stație mai îndepărtată astfel încât costul biletelor să nu depășească bugetul S și ora de plecare să fie mai târzie decât cea de sosire a trenului inițial. Dacă găsim o stație care să îndeplinească toate aceste condiții o reținem, iar la final o afișăm pe ultima găsită, dacă nu s-a găsit nicio stație afișăm mesajul: EXPEDITIE ESUATA :(.