

DESCRIERE SOLUȚII

PROBLEMA 1

Autor: student Dospra Cristian
Facultatea de Informatică, Universitatea București

NUME

Soluție O(N) – 100p

Prima observație este că obținem cea mai mare sumă pentru un șir de forma „AZAZA...AZ”. Diferența dintre oricare doi termeni este 25. Astfel putem determina numărul minim de caractere pe care îl va avea șirul final după următoarea formulă:

Termeni =	1	+	(N / 25)	+	(N % 25 ≠ 0)
Caracter	Primul alternativ		Câte pot pune se împarte la 25,		Dacă numărul nu mai pun încă o valoare de rest

Acum mai rămâne să îl facem minim lexicografic.

Strategia este următoarea:

1. Punem inițial caracterul ‚A’ (ca să fie minim)
2. Pentru fiecare caracter ce a mai rămas de pus, încerc să pun, pe rând, fiecare literă, pornind de la ‚A’ până la ‚Z’. Pentru fiecare literă, verific dacă pot obține cu restul de caractere rămase suma cerută. Dacă DA, atunci pun litera și trec la poziția următoare.
3. Ultima literă pusă va fi diferența rămasă până la suma cerută.

Observație: pentru a obține 100p, vom afișa literă cu literă la fiecare moment, fără a reține șirul (din lipsă de memorie). Se observă că singura literă din șir ce ne mai este utilă este ultima.

PROBLEMA 2

Autor: prof. Cristina Dragomirescu
Colegiul Național „Mihai Viteazul”, Ploiești

RUP

Pentru rezolvare se poate utiliza o variantă simplificată a generării partițiilor unei mulțimi, în care se pune condiția ca în vectorul soluție (cel care reține partiția) valorile să fie în ordine crescătoare.

Se ține cont și de numărul de apariții a numerelor de pe bilețele, cu ajutorul unui vector de frecvență.

PROBLEMA 3

Autor: prof. Iulia Lincan
Colegiul Național „Mihai Viteazul”, Ploiești

COD

În timpul citirii caracterelor din fișier se rețin permanent numerele formate din ultimele 3 cifre, ultimele 2 cifre și ultima cifră. Se determină inițial numărul de soluții obținute pentru prima cifră citită, primele două și primele trei cifre citite, având în vedere că:

- un număr de o cifră poate fi codificat cu literă mare numai dacă este nenul
- un număr de două cifre poate fi codificat cu literă mare numai dacă este cuprins între 10 și 26, sau cu literă mica doar dacă este mai mare sau egal cu 97
- un număr de trei cifre poate fi codificat cu literă mica doar dacă este cuprins între 100 și 122.

Apoi, se continuă citirea caracter cu caracter și dacă considerăm ca $f(n)$ este numărul variantelor de codificare pentru șirul format din primele n cifre, atunci el se poate obține ca sumă a variantelor de codificare pentru șirurile formate din primele $n-3$, $n-2$ respectiv $n-1$ cifre.

Astfel:

$f(n) = f(n-3) + f(n-2) + f(n-1)$, oricare ar fi $n \geq 4$, unde $f(n-3)$ se adună doar dacă numărul format din ultimele trei cifre respect condițiile de mai sus, $f(n-2)$ se adună doar dacă numărul format din ultimele două cifre respect condițiile de mai sus, iar $f(n-1)$ se adună la sumă doar dacă numărul format din ultima cifră este nenul.

Algoritmul lucrează doar cu variabile simple, este liniar iar complexitatea sa este data de numărul de cifre ale lui n .

PROBLEMA 4

Autor: prof. Alice Georgescu
Colegiul Național "Mihai Viteazul", Ploiești

ETICHETA

Soluția 1.

Se verifica toate relațiile de precedență în fiecare etichetă.

Considerând m și n lungimea celor 2 șiruri care se verifică, complexitatea verificării unei etichete este $O(m*n)$

Soluția 2.

Se vor șterge toate caracterele din etichetă care nu au corespondent în șirul ordinii culorilor critice. Apoi se vor elimina caracterele consecutive egale și se verifică egalitatea șirului rămas cu șirul ordinii culorilor critice (și din acesta vor trebui eliminate caracterele care nu au corespondent)

Considerând m și n lungimea celor 2 șiruri care se verifică, complexitatea verificării unei etichete este $O(m*n + n)$

Soluția 3 – 100 puncte

Se asociază fiecărui caracter un număr de ordine pe baza șirului ordinii culorilor critice și apoi se parcurge liniar șirul etichetei verificând pentru fiecare caracter dacă respectă această ordine

Considerând m și n lungimea celor 2 șiruri care se verifică, complexitatea verificării unei etichete este $O(n+m)$ dar se folosește memorie suplimentară (un vector pentru indicii de ordine al caracterelor)