Sursa: `sirag.c, sirag.cpp, sirag.pas`

Problema 3

sirag - descrierea soluţiei

1. Sa incepem mai intai cu solutia de 10 de puncte ($L = 1$) si sa vedem cum o putem extinde. Observam ca este suficient sa gasim o subsecventa care sa contina doar perle magice si perle de acelasi tip (P1).

Sa notam cu *w* sirul celor *N* perle.
Pentru aceasta calculam vectorul ajutator *A*, cu urmatoarea semnificatie:

A[i] = cel mai mare j, 1 <= j <= i, astfel incat w[j] != *, daca exista
        i, altfel

Nu este greu de vazut cum se poate calcula acest vector in O(N).
O data ce l-am calculat, avem nevoie de vectorul *T,* cu urmatoarea semnificatie:
T[i] = cea mai lunga lungime ale unei subsecvente ale sirului initial, ce se termina pe i, si respecta proprietatea P1.
Din nou, si *T* poate fi calculat in O(N).
Maximul valorilor din vector *T* va reprezenta solutia cautata.

2. Sa vedem acum cum putem obtine solutia de 100 de puncte, pentru orice valoare a lui L.

Vom folosi tot cei 2 vectori, *A* si *T*, numai ca in loc sa se refere la valori consecutive, se vor referi la valori din *L* in *L*.
Mai exact, vom imparti valorile sirului initial in *L* clase, in functie de restul impartirii la *L*. Acum, pentru fiecare dintre aceste *L* clase, vom calcula cei doi vectori numai in interiorul clasei respective, ca la solutia de 10 de puncte.
Avand vectorul *T* calculat, observam ca o subsecventa cautata este formata din *L* benzi (clase) consecutive.
Mai exact, lungimea maxima a unei astfel de subsecvente care se termina pe pozitia *i,* este minimul valorilor *T[i - L + 1 ... i]*.
Acum, problema s-a redus la determinarea minimului pe secvente consecutive de lungime *L*. Observam ca putem face aceasta determinare in timp liniar, folosind structura de date double-ended queue (deque). De asemenea, putem folosi un heap, pentru o solutie O (N log L). Astfel se incheie solutia.

Autor: Cătalin Ştefan Tiseanu
Colaboratori: Florin Manea, Adrian Diaconu