Problema pscfft — Autor: Tamio-Vesa Nakajima

Solutia de 100 puncte:

In rezolvarea acestei probleme sunt utile cateva observatii:

1) Notand M = $10^{10^{10^{100}}}$,pentru o pereche de indici (i, i+1) unde (i+1) este nedivizibil cu s, este mereu cazul ca $FFT(M, s)_{i+1}$=(1+$FFT(M, s)_i$) % s
2) Astfel, daca am stii ca sirul trebuie sa apara la o pozitie ce da un rest k != 0 la impartirea cu s, am putea extinde sirul la stanga si la dreapta pana la pozitii divizibile cu s.
3) Observatia cheie e ca un sir care are lungime multiplu de s, si care incepe la o pozitie multiplu de s, este valida daca si numai daca sirul format prin eliminarea valorilor la pozitii nedivizibile cu s ramane valid. Mai mult, daca sirul ce rezulta din eliminarea acestor elemente apare prima data la pozitia x, atunci sirul initial incepe la pozitia s * x
4) Orice indice i care nu satisface (v_i + 1) % s = v_{i+1} o vom numi o pozitie de split. In orice subsecventa a sirului FFT(M, s), oricare doua pozitii de split se afla la distanta multiplu de s, si trebuie sa apara un split macar la 2 * s + 1 pozitii.

Acestea sunt suficiente pentru descrierea unei solutii. Solutia se imparte in mai multe cazuri:

1) Pentru n > 2 *s, verificam daca exista un split. Daca nu exista, sirul de query nu apare ca subsecventa (conform (4)). Daca gasim una, atunci putem deduce (din (1)) valoarea modulo s a pozitie unde incepe sirul. Aplicam observatia (2) pentru a extinde sirul pana la un multiplu de s, si apoi aplicam observatia (3) pentru a reduce sirul curent la un sir de marime floor(n / s) + 2.
2) Altfel, daca exista un split, oricum putem aplica o solutie asemanatoare (aveti grija la cazul cu n = 2)
3) In rest, trebuie tratat separat cazul in care nu exista splituri si 2 * s >= n > s, cazul in care nu exista splituri si s >= n si cazul pentru n = 2 (in care orice procedura recursiva ca cea din (1) poate cicla aici). Aici sunt mai multe splituri posibile si se prefera cel care genereaza sirul [1, 1], apoi [2, 2], s.a.m.d. pana la [0, 0].

Exemplu: Pentru v = [2, 0, 1, 2, 1], s = 3, obsrvam ca intre pozitiile 3 si 4 exista un split. Astfel, stim ca daca v apare undeva in FFT(M, 3), el va aparea la o pozitie ce da restul 2 modulo 3. Putem deci spune ca daca v apare undeva, va apare ca substring al lui [0, 1, 2, 0, 1, 2, 1, 2, 0] (extinzand sirul pana la pozitii divizibile cu s). Gasim deci acest sir. Luand doar pozitiile divizibile cu s, putem reduce gasirea lui [**0**, 1, 2, **0**, 1, 2, **1**, 2, 0] la gasirea lui [0, 0, 1]. Aplicand aceeasi logica, se reduce la gasirea lui [**1**, 2, 0, **1**, 2, 0], ce se reduce la gasirea lui [1, 1], ce apare la pozitia 8 (se poate deduce formula general a pentru n = 2 — aveti grija doar la cazul v = [0, 0]).

Solutia de 90 de puncte:
Coincide in mare parte cu cea de 100 de puncte, doar ca aici se implementeaza fara formula cazul n = 2, si se reduce recursiv sirul [x, y] la un [(y+s-1)%s, y] prin transformarile [(x+1)%s, y], …, [(y+s-2)%s, y]

Solutia de 60 de puncte:
Coincide in mare parte cu cea de 100 de puncte, doar ca aici nu se trateaza separat cazul N <= 2 * s fara split-uri. Aici, se incearca efectiv toate resturile modulo s pe care le poate avea pozitia la care apare sirul, se aplica odata pasul de extindere si de reducere de la solutia de

100 de puncte, reducandu-se astfel sirul la un sir cu n maxim 2. Pentru acestea (n <= 2) se demonstreaza ca este necesar sa construiesti sirul doar pana la pozitia 2* s ^ s.

Solutia de 20 de puncte:
Se genereaza sirul pana la 2 * s ^ s * N si se foloseste algoritmul KMP pentru a afla rezultatul.