Solutia 1 (20 de puncte)

Se fixeaza x, locul de inceput al unei secvente, intre toate valoriel [1, N]. Pentru un x fixat, se itereaza y, locul de sfarsit al secventei, in intervalul [x, N] si se tine un contor cnt care numara cate valori apar de un numar par de ori. In cazul in care cnt este 0, adaugam 1 la solutie. Actualizarea valorii cnt se face folosind vectori de frecventa.

Complexitate totala: O(N^2)

Solutia 2 (100 de puncte)

Se imparte sirul in intervale de radical din N. Fiecare interval se rezolva ca in solutia 1, ducand la o complexitate de O(N √N) pentru toate intervalele. Pana acum au fost numarate secventele care incep si se termina strict in interioriul unui interval din cele radical din N intervale; astfel, raman de numarat secventele care incep intr-un interval si se termina in altul. Pentru a numara aceste secvente, se aplica urmatorii pasi:

1. Se itereaza, pe rand, fiecare interval.
2. Fie [a, b] intervalul curent; se vor numara in continuare secventele care incep in [a, b] si se termina in [b + 1, N].
3. Fie K numarul de valori comune din intervalele [a, b] si [b + 1, N]; se vor normaliza aceste K valori, in ordinea aparitiei in intervalul [b + 1, N], astfel: prima valoare care apare in intervalul [b + 1, N] va avea valoarea 0, a doua va avea valoarea 1, … , a K-a va avea valoarea K - 1; se normalizeaza si valorile necomune, atribuindu-se valori incepand cu K, pentru a evita eventualele conflicte.
4. Pentru fiecare sufix din intervalul [a, b] se calculeaza valoarea m, data de valoarea in baza 2 a numarului [b\_k-1 b\_k-2 … b\_1 b\_0] modulo M, unde bitul b\_i este 1 daca valoarea i apare de un numar impar de ori, 0 altfel, iar M este un numar prim; se numara intr-un hash “aparitiiSufix” de cate ori apare fiecare valoare m.
5. Se itereaza toate prefixele din intervalul [b + 1, N], incepand cu prefixul care se termina in N si terminand cu prefixul [b + 1, b + 1]; fie i capatul curent din dreapta al prefixui; pentru prefixul [b + 1, i], se calculeaza valoarea m, in mod analog pasului 4 si pentru valoarea m se calculeaza c = complement(m), unde complement(x) returneaza complementul binar al lui x; la solutie se adauga aparitiiSufix[c]. Daca la un moment dat una din cele K valori comune dispare din prefix, aceasta va fi exclusa din setul valorilor comune si se va repeta pasul 4, recalculand mastile tuturor sufixelor pentru restul de K - 1 valori comune; se observa ca pasul 4 va fi repetat de K ori, unde K este maxim sqrt(N), iar complexitatea pasului 4 este de asemenea sqrt(N), concluzionand astfel ca, pentru pasul 5, complexitatea este O(N).

Efectuand cei 5 pasi pentru fiecare interval din cele sqrt(N), complexitatea totala a algoritmului este O(N \* sqrt(N)).