Soluție Permgroup

Autor: Eric Stăvărache

Observatia care duce la rezolvarea acestei probleme este ca permutarile alese ca generator pot fi foarte particulare, si mai exact permutari care fac swap intre doua elemente (de exemplu 1 2 3 4 ... i - 1 j i + 1 .... j - 1 i j + 1 ... N)
si cele care fac shiftare circulara a intregului sir cu un K (K K + 1 ... N 1 2 3 4 .. K - 1).

Acum cand vrem sa obtinem o anumita permutare, putem privi problema ca pe o sortare. Asociem o ordine elmentelor in functie de cum vrem
sa fie in sirul final, si incercam sa ordonam folosind operatiile alese.

1) N <= 50, Cost <= 125.000

O implementare naiva ar fi sort sa folosim toate cele N * (N - 1) / 2 swapuri si in total N operatii. Pentru i de la 1 la N folosind fix unul din operatiile
de swap se poate aduce valoarea i de oriunde ar fi pe pozitia corecta. Costul ar fi N * (N - 1) / 2 * N in cel mai rau caz

2) N <= 100, Cost <= 40.000

Se poate implementa Bubble Sort. Pentru asta e nevoie doar de 2 permutari operatii cea swap intre pozitiile 1 si 2, si shiftare circulare cu 1 spre stanga.
Cum numarul de operatii e cel mult 3 * N * (N - 1) / 2 si se aleg 2 operatii costul este N * (N - 1) * 3 cel mult

3) N <= 100, Cost <= 20.000

In loc de bubble sort se pot folosi toate swapurile 1 X cu X de la 2 la N, si sa potriveasca pe rand valorile de la N la 1, pentru a potrivi o valoare
este necesar de cel mutl 2 swapuri (unul sa ducem valoarea pe prima pozitie, si unul sa o ducem pe pozitia potrivita). Numarul total de operatii este
2N^2 iar numarul de operatii este N - 1, facand costul maxim astfel N * (N - 1) * 2

4) N <= 200, Cost <= 39.999

Pentru acest test era doar necesar sa se gaseasca orice solutie care face mai putin N^2 operatii

5) N <= 300 si Cost <= 130000

Este o rafinare a solutiei de la 3). In loc sa folosim N - 1 operatii, putem folosi 2 logN operatii. logN operatii cu shiftare cu 1, 2, 4, s.a.m.d spre stanga

si swap intre 1 si 1 + 1, 1 si 1 + 2, 1 si 1 + 4 si asa mai departe. Acum daca voiam sa potrivim valoarea N ne-ar fi costat cel mult logN operatii sa ajungem cu ea pe prima pozitie
si cel mult 2logN pentru a o muta pe pozitia potrivita (daca distanta este D intre pozitia actuala si unde vrem sa fie, putem pentru fiecare bit din reprezentarea binara a lui D
sa facem swap urmat de shiftare cu acest bit). Numarul de operatii este aproximativ 3N * logN, iar numarul de operatii alese este 2logN.

6) N <= 600 si Cost <= 140000

Orice rafinare a algorimtului anterior ar aduce punctajul pe acest test.

7) Se pot folosi urmatoarele operatii: sqrt(N) - 1 swapuri de forma 1 + X (cu X de la 2 la sqrt(N)).
Inca [N / sqrt(N)] swapuri de forma 1 + K * sqrt(N) cu K de la 1 la [N / sqrt(N)].
Shiftare circulare cu 1 la stanga.
Shiftare circulare cu unu la stanga dar doar a elementelor sqrt(N) + 1, sqrt(N) + 2, ..., N
Shiftare circulara cu sqrt(N) la dreapat, dar doar a elementelor sqrt(N) + 1, sqrt(N) + 2, ..., N

Acum se incearca a se rezovla problema cate sqrt(N) elemente odata.in felul urmator:
- Toate elementele care trebuie sa se afle in blocul curent de marime sqrt(N) sunt puse pe pozitiile lor ca la 3)
- Apoi sa incearca toate pozitiile de forma 1 + K * sqrt(N) si se muta si ele pe pozitiile lor corecte
- Dupa ce se epuizeaza aceste pozitii, se roteste circular cu unu la stanga sirul de la sqrt(N) + 1, ..., N si se repeta pasul anterior in total de sqrt(N) ori
- Trebuie avut grija la elementul care vrea pe pozitia 1 (in functie de rotatia actuala). Acesta cand se gaseste trebuie asezat acolo
unde va fi sezat ultimul element gasit pe pozitii > sqrt(N) in rotatia actuala.
- Dupa ce se termina blocul curent, se roteste circular la dreapta sirul de la sqrt(N) + 1 la N (pentru a se readuce in forma initiala - elementele care se vor potrivi in blocul curent).
- Apoi se roteste circular la stanga tot sirul de sqrt(N) pentru a se sari la blocul urmator (eventual mai putin pentru ultimul bloc care nu e de marime sqrt(N))

In total se aleg 2sqrt(N) + 3 operatii si se fac 4N operatii pentur cost total aproximativ 8Nsqrt(N)

8) Algoritmul anterior poate fi optimizat observat ca rotirea circulara la stanga a intregului sir fie se face de sqrt(N) ori consecutiv (pentru aproape toate blocurile)
fie de un numar R de ori unde R e restul impartirii lui N la sqrt(N). Astfel se pot adauga aceste 2 operatii, iar acum numarul de operatii este 3N + sqrt(N), cu 2sqrt(N) + 4
operatii alese.