



Lexicografic

Constantinescu Andrei Costin

1. $K = \frac{N \cdot (N-1)}{2}$

În acest caz, operațiile se pot folosi pentru a sorta perfect numerele. Sortarea poate fi executată prin orice metodă, spre exemplu utilizând funcția *sort* din STL.

2. $K = 1$

Există o singură operație. Se observă că pentru a face această operație utilă, ea trebuie aplicată pe două poziții i și $i + 1$ astfel încât $v_i > v_{i+1}$. Pentru a obține șirul minim lexicografic trebuie aleasă poziția i minimă cu această proprietate.

3. $N = 2\,000$

Se va parcurge șirul în ordine crescătoare a indicilor. Pentru fiecare poziție i de la 1 la N se va alege cel mai mic element dintre $v_i, v_{i+1}, \dots, v_{\min(n, i+K)}$ și se va aduce pe poziția i prin operații succesive de interschimbare. În caz de egalitate se va alege primul dintre aceste elemente. La sfârșitul pasului curent se va scădea din K numărul de operații efectuate.

Se poate demonstra că acest algoritm duce la soluția optimă. Argumentul central este urmatorul: dacă un element v_i oarecare trebuie adus mai la stânga decât alt element v_j în soluția finală, atunci v_i merită mutat până la poziția lui finală înaintea lui v_j .

Complexitatea acestei soluții este $O(N^2)$.

4. $N = 250\,000$

Soluția anterioară se poate optimiza la $O(N \log N)$ folosind structuri de date eficiente pentru a îmbunătăți căutarea elementului care trebuie adus pe poziția i . Spre exemplu, se poate menține un arbore de intervale care stochează minimul pe interval și numărul de elemente încă nemutate din interval. Cu ajutorul parametrului din urmă și al lui K , se caută minimul în intervalul $[1, \min(n, K + 1)]$ de **numere nemutate**. Se șterge elementul din structură, se adaugă la finalul unui vector în care se ține soluția, se scade K cu numărul de operații necesare și se trece la găsirea următorului element.