Qswap

2021, Baraj 2, Seniori

Putem considera, vectorul ca fiind o listă de puncte într-un plan, unde punctul (i, p[i]) este punctul reprezentând al i-lea număr din vector.

Se poate observa că numărul de operații necesare pentru sortarea unui șir este egal cu numărul de inversiuni prezente în șir. O inversiune este definită ca fiind două poziții i < j astfel încât p[i] > p[j]. Astfel, dorim ca operația specială să reducă numărul de inversiuni cât mai mult.

Se poate observa cu ușurință că nu este optim să schimbăm poziţiile i și j dacă p[i] <= p[j], așa că ne vom ocupa doar de celălalt caz.

Într-un schimb, vom considera dreptunghiul cu colţurile (i, p[i]) si (j,p[j]), astfel încât numărul total de inversiuni salvate este x + y - 1, unde x e numărul de puncte din interiorul dreptunghiului și y e numărul de puncte din interiorul sau de pe laturile dreptunghiului.

Pentru o poziţie i, putem găsi o submulțime de candidaţi strict crescătoare, așa că se poate demonstra că dacă r e mai mic ca s, valoarea optimă a lui j pentru r e mai mică decât valoarea optimă a lui j pentru s. Astfel, putem folosi optimizarea divide and conquer pentru a rezolva problema în O(nlog^2n).