Snpid – descrierea solutiei

Mugurel Ionut Andreica – Universitatea Politehnica din Bucuresti

Solutia 1 - O(N*log(N)) – Mugurel Ionut Andreica - 100 de puncte

Se considera punctele in ordine inversa (de la cel cu x=N catre cel cu x=1). Vom pastra intr-o structura de date toate coordonatele y ale punctelor peste care am trecut pana acum (cand suntem la punctul i, in structura exista coordonatele y ale punctelor i+1, …, N). Vom calcula numarul P de puncte din structura avand coordonata y > y(i) (i=punctul curent). Valoarea SNPID(i) este egala cu 0 + 1 + … + (P-1) minus numarul de inversiuni dintre coordonatele x ale punctelor avand coordinate mai mari decat y(i). Dupa ce terminam cu punctul i il introducem in structura de date.

Putem pastra numarul de inversiuni utilizand un arbore de intervale cu urmatoarele tipuri de operatii:

- query(A) : intoarce numarul de inversiuni dintre toate punctele cu coordonate y >= A
- update(Q, A) : creste numarul de inversiuni pentru pozitiile active din intervalul [1,A], in felul urmator: pentru prima pozitie activa numarul de inversiuni creste cu Q, pentru a doua creste cu Q-1, s.a.m.d.

In fiecare nod al arborelui de intervale vom tine 2 valori: Qsum si Ksum, avand semnificatia urmatoare: pentru prima coordonata activa din intervalul corespunzator nodului, numarul de inversiuni trebuie sa creasca cu Qsum, pentru a doua coordonata activa numarul de inversiuni va creste cu Qsum – Ksum pentru a treia coordonata activa cu Qsum – 2*Ksum s.a.m.d.

Un update va incrementa valorile Qsum si Ksum din nodurile care "acopera" intervalul [1,A] (Ksum e incrementat cu 1, Qsum e incrementat cu o valoare diferita in functie de intervalul actualizat). O interogare va urca in sus arbore si va mentine a cata valoare activa R este in cadrul intervalului corespunzator nodului curent; pe baza acestor informatii, se adauga Qsum – (R-1) * Ksum la rezultatul interogarii (Qsum si Ksum corespund nodului curent).

Dupa ce am calculat raspunsul pentru un punct i, vom apela *update(N-i-P, y(i)-1)* si apoi vom introduce coordonata y(i) in arbore. Pentru toate nodurile de pe drumul de la y(i) la radacina va trebui sa "resetam" valorile Qsum si Ksum (caci altfel s-ar aplica si asupra lui y(i), ceea ce ar fi incorect). Inainte de restare, aceste valori vor fi "impinse" catre nodurile de o parte si de alta a drumului de la y(i) la radacina arborelui de intervale. In plus, inainte de introducerea lui y(i) in arbore, vom salva separat care este numarul de inversiuni curente de deasupra pozitiei y(i) (acest numar ar fi pierdut dupa "resetare" si ne mai trebuie la interogari ulterioare).

Solutia 2 - O(N*log(N)) – Marius Stroe - 100 de puncte

Presupunem ca punctele au fost translatate astfel incat cerinta coltului "stanga jos" devine "dreapta-sus".

Punctele vor fi procesate crescator dupa ordonata, Oy. In plus, pentru un punct B definim S(x(b), y(b)) ca fiind numarul punctelor A cu x(a) < x(b), adica punctele aflat "la stanga" punctului curent B.

Solutia naiva, pentru fiecare punct B, consta in a select fiecare punct A, cu x(a) < x(b), si de a numara toate punctele din interiorul dreptunghiurilor cu ajutorul sumelor partiale S(). Formula este:

Cnt(B) = Suma(A: x(a) < x(b)) [S(x(b), y(b)) – S(x(a), y(b)) – S(x(b), y(a)) + S(x(a), y(a))]
⇔
Cnt(B) = n * S(x(b), y(b)) + Suma(A: x(a) < x(b)) S(x(a), y(a)) - Suma(A: x(a) < x(b)) [S(x(a), y(b)) – S(x(b), y(a))], unde n reprezinta numarul punctelor la stanga punctului B.
⇔
Cnt(B) = n^2 + Suma(A: x(a) < x(b)) S(x(a), y(a)) - Suma(A: x(a) < x(b)) [S(x(a), y(b)) – S(x(b), y(a))]

Pentru a calcula S(x(b), y(b)) (sau n), vom folosi un arbore indexat binar in care adaugam 1 la indexul x(c) fiecarui punct nou C.

Termenul urmator se poate obtine in mod similar dintr-un arbore indexat binar daca fiecarui punct nou C, adaugam S(x(c), y(c)) indexului x(c).

Ultimii doi termeni se obtin prin a fixa capatul dreptunghiului in fiecare din punctele la stanga lui B. Acest lucru inseamna ca S(x(a), y(b)) este 1, 2, 3, .., n – 1, pentru fiecare punct A cu x(a) < x(b). Analog si pentru S(x(b), y(a)).

Astfel, formula devine:
Cnt(B) = n^2 – n * (n – 1) + Suma(A: x(a) < x(b)) S(x(a), y(a))
⇔
Cnt(B) = n + Suma(A: x(a) < x(b)) S(x(a), y(a))

Solutia 3 – complexitate variabila – Adrian Panaete, Mugurel Ionut Andreica – 40-60 de puncte

Putem introduce toate punctele intr-o structura de date 2D (quad-tree, kd-tree, arbori de intervale 2D, etc.). Luam pe rand fiecare punct si calculam numarul de puncte P aflate in cadranul de Nord-Est relativ la coordonata punctului. Apoi facem un update cu valoarea P pe intreg cadranul de Sud-Vest relativ la coordonata punctului (incrementam cu P o variabila din fiecare nod al structurii ce “acopera” cadranul Sud-Vest). Rezultatul pentru un punct i se va obtine pornind din punctul i si trecand prin toate nodurile structurii de date 2D ce contin nodul i si adunand valorile X din acele noduri (fiecare nod al unei astfel de structuri corespunde unei zone dreptunghiulare din plan). De ex., in cazul arborilor de intervale 2D, exista $O(\log^2(N))$ astfel de noduri, in cazul kd-tree este O(sqrt(N)) astfel de noduri, etc. (un numar similar de noduri “acopera” si o zona dreptunghiulare in cazul update-urilor).

Per total, complexitatea obtinuta este O(N*sqrt(N)) in cazul kd-tree, $O(N*\log^2(N))$ in cazul arborilor de intervale 2D, etc.

Solutia 4 - $O(N^2*\log(N))$ – 30-40 de puncte

Se considera fiecare dreptunghi si se efectueaza o interogare intr-o structura de date pentru a calcula numarul de puncte din interiorul dreptunghiului. Daca fixam coltul stanga-jos si apoi baleiem punctele spre dreapta se poate utiliza o structura 1D de tipul AIB/arbori de intervale/etc.

Solutia 5 - $O(N^3)$ – 20 de puncte

Se considera fiecare dreptunghi si se calculeaza in timp O(N) numarul de puncte din interiorul sau.