"Convex hull trick"

Ülesanne 1. On antud N funktsiooni kujul $f_1(x), f_2(x), \ldots, f_N(x)$, igaüks neist kujul $f_i(x) = a_i x + b_i$. Antakse Q päringut, igaüks neist kujul:

- Antakse arv x. Leia $\max_i f_i(x)$ ehk suurim arvude $f_1(x), f_2(x), \ldots, f_N(x)$ seast.

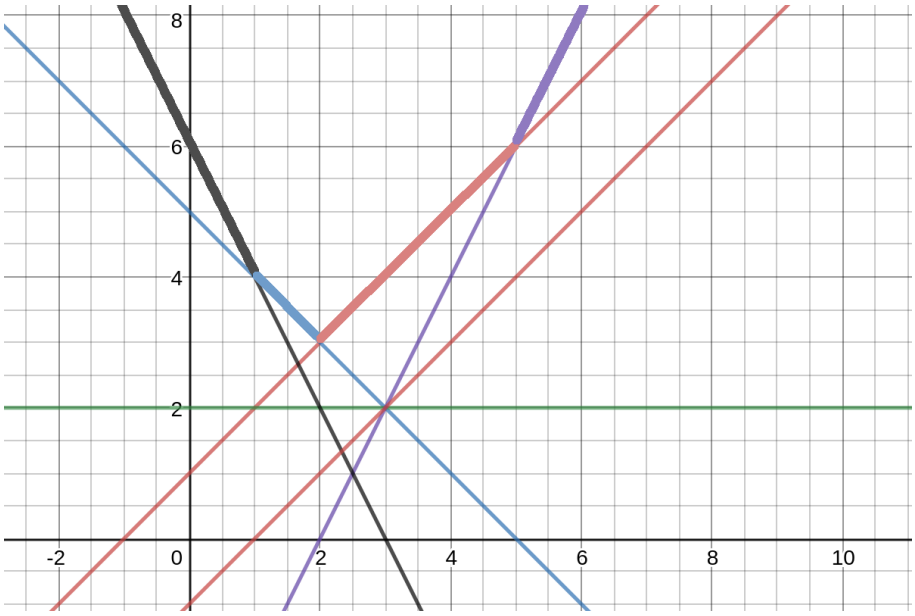
Vastata kõikidele päringutele. $1 \leqslant N, Q \leqslant 10^5$.

Näide 1. Näiteks võivad meil olla sellised funktsioonid.

$$f_1(x) = x + 1;$$
$$f_2(x) = -x + 5;$$
$$f_3(x) = 2;$$
$$f_4(x) = 2x - 4;$$
$$f_5(x) = -2x + 6;$$
$$f_6(x) = x - 1.$$

Edaspidise arutelu jooksul kasutame seda näidet.

Iga funktsiooni kohta võime küsida küsimuse "missuguste x väärtuste korral on selle funktsiooni väärtused kõikidest teistest suuremad". Ütleme, et selles piirkonnas antud funktsioon *domineerib*.



Joonis 1: Funktsioonide domineerimispiirkonnad

Näide 2. Näite 1 funktsioonide seas (mis on näidatud joonisel 1):

- $f_1(x) = x + 1$ domineerib intervallis $[3, 5]$;
- $f_2(x) = -x + 5$ domineerib intervallis $[1, 2]$;
- $f_3(x) = 0x + 2$ ei domineeri kuskil;
- $f_4(x) = 2x - 4$ domineerib intervallis $[5, \infty]$;

- $f_5(x) = -2x + 6$ ei domineerib intervallis $[-\infty, 1]$;
- $f_6(x) = x - 1$ ei domineeri kuskil.

Meie eesmärk on konstrueerida andmestruktuur, mille abil saab kiiresti kontrollida, missuguse funktsiooni "domineerimispiirkonda" mingi etteantud x jääb. Kõige lihtsam viis oleks konstrueerida massiiv, kus on domineerimispiirkonnad õiges järjestuses kirja pandud. Meie näite korral võiks see välja näha näiteks nii:

$$[\langle -\infty, f_5 \rangle, \langle 1, f_2 \rangle, \langle 3, f_1 \rangle, \langle 5, f_4 \rangle].$$

See on massiiv, mille elemendid on paarid: paari parempoolne element on funktsioon ja vasakpoolne element on vähim selline x, mille korral antud funktsioon domineerib. Kui meil oleks selline massiiv, siis võime etteantud x korral lihtsalt kahendotsinguga otsida selle funktsiooni, mille domineerimispiirkonnas x asub, mille abil saame vastata päringutele.

Seega uurime, kuidas konstrueerida sellist domineerimispiirkondade massiivi keerukusega $\mathcal{O}(N)$. Suhteliselt lihtne on veenduda, et kehtib järgmine tähelepanek:

Tähelepanek 1. Funktsioonid domineerivad tõusude järjekorras, s.t. mida väikesem x, seda väikesema tõusuga funktsiooni domineerimispiirkonnas ta on.

Sorteerime funktsioonid tõusu järgi kasvavalt. Lisaks, kui meie funktsioonide hulgas on paralleelseid (s.t. sama tõusuga) funktsioone, siis jätame alles ainult kõige suurema.

Olgu $f_1, f_2, \ldots, f_N$ sorteeritud tõusu järgi kasvavalt. Kasutame järgmist algoritmi.

- cht $= [\langle -\infty, f_1 \rangle]$;
- iga $i = 2, 3, \ldots, N$ kohta:
  - niikaua, kuni meet$(f, g) < c$, kus $\langle c, g \rangle$ on cht viimane element:
    - eemalda cht viimane element massiivist;
  - lisa massiivi paar $\langle \text{meet}(f, g), f \rangle$, kus $\langle c, g \rangle$ on cht viimane element.

Siin meet(f, g) on see (üheselt määratud) x, et $f(x) = g(x)$. Keerukus $\mathcal{O}(N)$.

See võte on väga paindlik. Näiteks ei pea funktsioonid tingimata olema sirged. Näiteks on järgmine ülesanne lahendatav täpselt samasuguse võttega:

Ülesanne 2. Tasandil on antud N pitsarestorani. Antakse Q päringut kujul:

- Antakse arv x. Leida punktile $(x, 0)$ lähim pitsarestoran.

Asugu mõni pitsarestoran punktis (x_i, y_i). Tema kaugus punktist x on siis $(x_i - x)^2 + y_i^2 = x^2 - 2x_i x + (x_i^2 + y_i^2)$, mis on parabool. Aga lahendus on väga sarnane: paraboolid domineerivad haripunkti asukoha järjekorras, kõik muu jääb samaks.

Samuti on võimalik teha võttest nn. "dünaamiline" versioon:

Ülesanne 3. On antud N funktsiooni kujul $f_1(x), f_2(x), \ldots, f_N(x)$, igaüks neist kujul $f_i(x) = a_i x + b_i$. Antakse Q päringut, igaüks neist kujul:

- Antakse arv x. Leia $\max_i f_i(x)$.
- Antakse arvud a ja b. Lisada funktsioonid sekka uus funktsioon $ax + b$.

Vastata kõikidele päringutele. $1 \leqslant N, Q \leqslant 10^5$.

Nüüd tuleb domineerimispiirkondade massiivis teha uuendusi. Kui funktsiooni sinna lisada, tuleb leida temale õige koht, ta sellesse kohta juurde lisada ja seejärel arvutada tema naaberfunktsioonide uued domineerimispiirkonnad (neid võibolla ka domineerimismassiivist eemaldades).

Rakendusi

Sageli on seda võtet võimalik kasutada DP lahenduste optimeerimiseks. Näiteks võib olla, et mingil ülesandel on DP lahendus, mille keerukus on $\mathcal{O}(N^2)$. Samas dünaamilise programmeerimise valem on selline, kus arvutustes võetakse samamoodi näiteks mingite lineaarfunktsioonide maksimume nii, et meie trikiga on seda võimalik optimeerida nii, et keerukus oleks näiteks $\mathcal{O}(N \log N)$.

Ülesanne 4. On N puud, kõrgustega vastavalt $1 = a_1 < a_2 < a_3 < \cdots < a_N$ meetrit, ning arvud $b_1 > b_2 > b_3 > \cdots > b_N = 0$. Meie eesmärk on võtta maha kõik puud.

Selle jaoks on meil mootorsaag, mis suudab maha võtta ühe meetri korraga ja vajab seejärel taaslaadimist. Mootorsae laadimise hind on b_j, kus j on kõrgeima täielikult maha võetud puu indeks.

Mootorsaag on alguses laetud. Leia odavaim viis kõik puud maha võtta. $1 \leqslant N \leqslant 10^5$.

Tähelepanek 2. Enne puu N maha võtmist peaks maha võetavate puude jada olema kasvav.

Tõepoolest, kujutame ette, et võtame maha puu i ja seejärel puu j, kusjuures $i > j$ ja puu N ei ole veel maha võetud. Siis puu j maha võtmine kulutab mingi hulga raha, aga ei tee teiste puude maha võtmist odavamaks. Kui võtaksime puu j maha pärast puu N maha võtmist, kulutaksime vähem raha.

Järelikult peame valima indeksid $1 = i_1 < i_2 < \cdots < i_k = N$ nii, et nende puude maha võtmine selles järjekorras oleks võimalikult odav. Tähistagu $\mathrm{dp}[t]$ odavaimat viisi valida arvud $1 = i_1 < i_2 < \cdots < i_k = t$ ja võtta maha vastavad puud selles järjekorras. Siis

$$\begin{aligned} \mathrm{dp}[1] &= 1; \\ \mathrm{dp}[t] &= \min_{1 \leqslant k < t} \mathrm{dp}[k] + b_k a_t. \end{aligned}$$

Kuidas arvutada kiiresti $\mathrm{dp}[t]$? Paneme tähele, et see on lihtsalt lineaarfunktsioonide $b_1 x + \mathrm{dp}[1], b_2 x + \mathrm{dp}[2], \cdots, b_{t-1} x + \mathrm{dp}[t-1]$ miinimum punktis $x = a_t$. Järelikult, hoiame meeles "domineerimismassiivi", arvutame iga t korral $\mathrm{dp}[t]$ kahendotsinguga ja siis lisame sinna sirge $\mathrm{dp}[t] + b_t x$. Seejuures selle tõus on väikesem, kui kõikidel eelmistel sirgetel. Nii lahendame ülesande keerukusega $\mathcal{O}(N \log N)$.