Намиране на лицето на обединението на правоъгълници

Разглеждаме множество от N правоъгълника, страните на които са успоредни на осите на правоъгълна координатна система. Търсим лицето на заетата от правоъгълниците част от равнината.

Преди всичко да обърнем внимание, че търсеното лице в общия случай не е сбор от лицата на правоъгълниците, защото всеки правоъгълник може напълно или отчасти да покрива някои от останалите. Например, ако един от правоъгълниците съдържа всички останали, търсеното лице е неговото. Изобщо, трябва да избегнем „преброяването“ на която и да е част от лицето повече от веднъж.

Подходяща идея как да стане това ни дава решаването на подобна, но по-проста задача – вместо правоъгълници разглеждаме отсечки върху числовата ос, т. е. числови интервали, и търсим дължината на заетата от интервалите част от оста. Иначе казано, подобно на първоначално поставената задача, търсим мярката на обединението – в случая на отсечки върху права вместо на правоъгълници в равнината – и мярката е дължина вместо лице.

Нека числовите интервали са (u_i, v_i) за $i = 1, \ldots, N$. Да подредим в ненамаляващ ред всички числа u_i и v_i: получаваме редица от $2N$ числа x_i, за всяко от които е отбелязано дали то е ляв или десен край на интервал.

Всеки две съседни в редицата числа образуват отрязък, който или изцяло се съдържа в някои от интервалите (u_i, v_i), или е изцяло извън всички тях. Това позволява да решим задачата по следния начин: обхождаме поред числата x_i, $i = 2, \ldots, 2N$, като добавяме дължината $x_i - x_{i-1}$ на всеки попадащ в поне един от интервалите (u_j, v_j) отрязък (x_{i-1}, x_i) към търсената стойност, отначало 0. За да знаем дали поредният отрязък попада в интервал, поддържаме брояч, който се увеличава с 1 при преминаване през ляв край и намалява с 1 при преминаване през десен край на интервал. Така броячът е положителен в онези и само в онези участъци, които са покрити от поне един интервал.

Описаният процес се представя така – броячът е c, а намираната дължина – s:

$s := 0$; $c := 1$
за $i := 2$ **до** $2N$:
  ако $c > 0$: $s \mathrel{+}:= x_i - x_{i-1}$
  ако x_i е лява граница: $c \mathrel{+}:= 1$
  иначе: $c \mathrel{-}:= 1$

Възможно най-малката времева сложност, с която решаваме задачата, е пропорционална на $N \log N$ – тази на подреждането, ако то се изпълнява оптимално. Горният процес е по-бърз от подреждането – едва линеен относно N.

Сега да се върнем на основната задача – за лицето на множество от правоъгълници. Всеки правоъгълник се задава с двойка интервали – един за хоризонталните и един за вертикалните му граници. Да подредим и тук хоризонталните граници – получаваме $2N$ числа в ненамаляващ ред. Вертикалните прави с тези абсциси образуват в равнината $2N - 1$ на брой подредени отляво надясно вертикални ивици. Дадена ивица е празна, ако двете ограничаващи я абсцици са равни.

За коя да е непразна ивица и вътрешността на кой да е правоъгълник те или нямат общи точки, или в ивицата попада правоъгълна част от правоъгълника (или целият той). При това хоризонталните граници на тази част са тези на ивицата, а вертикалните са тези на правоъгълника. Когато лявата граница на даден правоъгълник се окаже лява граница на някоя ивица, тази и възможно известен брой следващи ивици – включително до онази, на която дясна граница е дясната граница на правоъгълника – са тъкмо тези ивици, в които има части от правоъгълника.

Търсеното лице на обединението на правоъгълниците е сбор от частите на обединението във всяка от ивиците. За да намерим лицето на обединението, можем да обходим ивиците една по една, като всеки път добавяме намереното за текущата ивица лице – прилагаме метода на метенето с вертикална права, която посещава границите на ивиците. За да става

това ефективно по време, имаме нужда от удобна структура за представяне на съдържанието на текущата ивица. Структурата трябва да позволява относително бързо да се изпълняват следните две действия:

- включване на правоъгълник в ивицата и изключване от нея;
- намиране на лицето на обединението на интервалите от вертикалните граници на попадащите в ивицата правоъгълници.

Второто действие по същество съвпада с по-простата, едномерна задача, решение на която приведохме по-горе. Обаче, ако използваме и тук точно същото решение, тъй като ще трябва то да се възпроизведе за всяка ивица, получаваната обща сложност би била с порядък $N^2 \log N$, и то без да броим включването и изключването на правоъгълници. Истински добрата структура за представяне на ивица позволява при намиране на лицето на обединението на интервалите в ивица *да избегнем подреждането* и с това да постигнем по-високо бързодействие. Тя е особен вид двоично дърво, позволяващо ефективно образуване, изменяне и други действия с подмножества на предварително известно множество от интервали. Нарича се *отсечково дърво* (различно от подобното по име и предназначение *интервално дърво*). По-нататък описваме именно него.

Отсечково дърво

Избирането на цели неотрицателни числа $a < b$ от множеството $0, 1, \ldots, 2^k$, където $k \geq 0$, позволява да образуваме всички възможни интервали $[a, b]$, които се съдържат в $[0, 2^k]$, както и самия него. Отсечковото дърво (ОД) от ред k е пълно (без празнини) двоично дърво с $k+1$ слоя, в което всеки възел отговаря на някой от тези интервали. Като дадем на слоевете номера от 0 до k, на корена на дървото (в слой 0) отговаря най-големият интервал $[0, 2^k]$. На левия и десния наследници на корена, в слой 1, отговарят съответно половините $[0, 2^{k-1}]$ и $[2^{k-1}, 2^k]$ от кореновия интервал. По същия начин на техните наследници отговарят половините от тези половини и т. н. – изобщо на възлите в кой да е слой i отговарят интервали с дължина 2^{k-i}, които в ред отляво надясно са строго последователни и слети образуват $[0, 2^k]$. На възлите от последния слой k отговарят интервалите с най-малката дължина 1.

На всеки възел в дървото отговаря по точно един интервал и всички тези интервали са различни, но има интервали, които не се образуват с описания процес на разполовяване и затова те нямат единствен съответен възел. Такъв интервал еднозначно съответства на някакво множество от два или повече възела. Например в дърво от ред 4 интервалът $[5, 12]$, тъй като се представя във вида $[5, 6] \cup [6, 8] \cup [8, 12]$, съответства на възлите, отговарящи на тези три интервала. Така за ОД от кой да е ред k е налице взаимноеднозначно съответствие между подинтервалите на $[0, 2^k]$ и някои подмножества с един или повече членове на възлите на дървото.

Намирането на отговарящите на даден интервал $[a, b] \subseteq [0, 2^k]$ възли на дървото или съответните им интервали може да стане с процедурата

```
проц намери ([a, b], n)
    p, q := краищата на интервала, съответен на n
    ако [p, q] ⊆ [a, b]:
        съобщи [p, q]
    иначе:
        m := (p+q)/2
        ако a < m: намери ([a, b], n<)
        ако m < b: намери ([a, b], n>)
```

като я повикаме, давайки за стойност на n корена на дървото. С $n_<$ и $n_>$ в повикванията на процедурата към себе си означаваме левия и десен наследници на възела n.

Сложността на работата на процедурата може да се измери по броя на направените повиквания за даден интервал $[a, b]$. При кое да е повикване е вярно $[a, b] \cap [p, q] \neq \varnothing$ и следователно или $[p, q] \subseteq [a, b]$, или $[p, q]$ съдържа поне едното от a и b. От строежа на дървото е ясно, че в кой да е негов слой всеки от двата случая възниква при не повече от два възела, и тъй като слоевете са $k+1$, общият брой повиквания на процедурата е по-малък от $4k$.

Вместо просто да намираме и съобщаваме съответните на $[a,b]$ интервали от възлите на дървото, има смисъл да съхраняваме информация за наличието или отсъствието на интервалите $[a,b]$ в самото дърво. За тази цел във всеки възел n на дървото поставяме по един брояч n_c, чиято роля е като на брояча в едномерната задача: той отразява колко интервала, в чието представяне участва съответният на възела интервал, присъстват в дървото. Добавянето или премахването на интервал към или от запомнените в дървото се състои в увеличаване или намаляване с 1 на броячите в съответните възли.

Така преправяме горната процедура в нова, която наричаме *вкл-изкл* и повикваме със стойност 1 или -1 на параметъра δ съответно при добавяне (включване) и премахване (изключване) на интервал от дървото. В процедурата *вкл-изкл* отчитаме и следното. Попрактично е да не пресмятаме p и q за всеки отделен възел n, а да използваме това, че интервалите, съответни на наследниците на даден възел, са половините на интервала, съответен на този възел. Така при всяко първоначално повикване на *вкл-изкл* – когато стойност на n е коренът на дървото – променливите p и q, вече също параметри на процедурата, получават стойности 0 и 2^k, а по-нататък, при повикванията на тази процедура към себе си, тя избира за p и q очевидните нови стойности.

проц *вкл-изкл* $(n,\ [a,b],\ [p,q],\ \delta)$
  $m\ :=\ (p+q)/2$
  ако $[p,q] \subseteq [a,b]$:
    $n_c\ +:=\ \delta$
  иначе:
    ако $a < m$: *вкл-изкл* $(n_<,\ [a,b],\ [p,m],\ \delta)$
    ако $m < b$: *вкл-изкл* $(n_>,\ [a,b],\ [m,q],\ \delta)$

Процедурата *вкл-изкл* дава възможност да добавим към или премахнем от ОД от ред k кой да е интервал с целочислени граници $[a,b] \subseteq [0,2^k]$ и сложността на такова действие е $\sim k$. Ако във възлите на дървото освен броячите се съхранява още някаква информация, могат да се добавят действия, които обобщават тази информация от крайните възли нагоре към корена. Подходящо място за такива действия е краят на процедурата, понеже там чрез рекурсивните повиквания обобщаването вече е направено за поддърветата и така то може да се извърши за текущия възел. Точно така постъпваме, за да използваме отсечково дърво за намиране на лицето на обединението на правоъгълници.

Прилагане на отсечково дърво за решаване на задачата

Да подредим в ненамаляващ ред числата, които в координатната система задават горните и долни граници на правоъгълниците. Нека k е най-малкото число, за което $2^k \geq 2N-1$. Подредените числа, $2N$ на брой, образуват редица $y_0, y_1, \ldots, y_k$, която е допълнена с фиктивни членове, ако това е нужно, така че да има посочената дължина. За индексите $0, 1, \ldots, k$ на редицата образуваме отсечково дърво от ред k.

Всеки възел n на дървото съдържа две стойности, и двете първоначално 0. Едната е броячът n_c, както описахме. Другата е число n_{cov} – покритие на n – което съдържа дължината на частта от съответния на n интервал, заета от включените в дървото в даден момент интервали $[i,j]$, където y_i и y_j са долната и горна граници на правоъгълник, от който има част в текущата вертикална ивица.

Допълваме *вкл-изкл* с обновяване на стойностите n_{cov}, което добавяме в самия край на процедурата:

$$n_{cov}\ :=\ \begin{array}{ll} \textbf{ако}\ n_c > 0\text{:} & y_q - y_p \\ \textbf{иначе ако}\ n \text{ не е краен:} & (n_<)_{cov} + (n_>)_{cov} \\ \textbf{иначе}\text{:} & 0 \end{array}$$

Главната част на решението на задачата е пробягване на вертикалните ивици, т. е. на редицата $x_1, x_2, \ldots, x_{2N}$ от подредените в ненамаляващ ред леви и десни граници на правоъгълниците. За всяка ивица към търсеното лице s се прибавя произведението на нейната ширина $x_i - x_{i-1}$ с дължината *корен*$_{cov}$ на обединението на интервалите от вертикалните граници на попадащите в ивицата части от правоъгълници. Накрая, индексите $a < b$ на долната и горна граници на правоъгълника, на който хоризонтална граница е дясната граница

x_i на ивицата (a и b са индекси на числа от редицата $y_0, y_1, \ldots, y_{2N-1}$), задават интервал, който се добавя към или отнема от отсечковото дърво според това, дали x_i е лява или дясна граница на правоъгълника.

Полученият алгоритъм има следния вид:

- Подреждане поотделно на хоризонталните и на вертикалните граници на правоъгълниците
- Образуване на ОД от ред k; за всеки възел n полагаме $n_c := 0$ и $n_{cov} := 0$
- $s := 0$
- **за** $i := 2$ **до** $2N$:
  s $+:=$ $(x_i - x_{i-1})$ *корен*$_{cov}$
  a, b $=$ индекси на верт. граници на правоъгълника, на който хор. граница е x_i
  вкл-изкл (*корен*, $[a, b]$, $[0, 2^k]$, **ако** x_i е лява граница: 1 **иначе**: -1)

Тъй като отсечковото дърво е пълно двоично дърво, то се представя по възможно най-простия начин: като редица от възлите си, изброени слой по слой, всеки слой отляво надясно. Така възлите на дървото имат последователни номера $0, 1, 2, \ldots$, като 0 отговаря на корена на дървото. Тогава в процедурата *вкл-изкл* параметърът n е такова число, 0 за всяко начално повикване, а за левия и десен наследници на n е вярно $n_< = 2n+1$ и $n_> = 2n+2$.

Редицата от възлите на дървото условно има $2^{k+1}-1$ члена, но тези от последния слой след първите $2N-1$, тъй като не се използват и са в края на редицата, може изобщо да не съществуват. Това може да спести до около четвърт от използваното за представяне на дървото пространство.

Предвид времевата сложност на процедурата *вкл-изкл* – като тази на *намери* – и това, че подреждането на хоризонталните и на вертикалните координати, извършено оптимално, отнема време $\sim N \log N$, такова е и времето, за което решава задачата алгоритъмът като цяло. Забележително е, че решението е от същия порядък на сложност като това на едномерната задача.

Подходът на метене с права, съчетан с използване на отсечково дърво, може да се приложи за решаване и на други задачи за множества от правоъгълници с взаимно успоредно-отвесни страни, например намиране на контура на обединението (една или повече затворени начупени линии) или периметъра (дължината на контура) или такива обекти за обединение с „дупки“.

Б. Банчев
1.2026