Анализ на задача „Интервали“

Автор: Виктор Кожухаров

Формално, задачата е следната: дадени са N скрити отворени интервала (L_i, R_i). С една заявка избираме непразно множество от интервали и научаваме дължината на тяхното сечение. Трябва да намерим възможно най-много интервали, които имат обща точка, като зададем възможно най-малко заявки. Пълен брой точки се получава, ако заявките не са повече от $N(N+1)/2$.

Ще видим, че е достатъчно да питаме за всяка двойка интервали и че в най-лошия случай по-малко заявки не стигат. Основната трудност е как от отговорите за двойките да намерим търсените интервали, без да знаем къде точно се намират.

Навсякъде ще означаваме с $d_i = R_i - L_i$ дължината на интервала i, а с $q(i, j)$ – отговора на заявката за двойката $\{i, j\}$.

1 Подзадача 1: $N \leq 10$, за 9 точки

Тук броят на заявките няма значение, стига отговорът да е верен. Обхождаме всички $2^N - 1$ непразни подмножества на интервалите и за всяко задаваме по една заявка. Връщаме най-голямото подмножество, за което отговорът е положителен. Това са най-много 1023 заявки, а сложността е $O(2^N \cdot N)$.

2 Подзадача 2: $N \leq 100$, за 61 точки

Кога няколко интервала имат обща точка. Сечението на няколко интервала е интервалът $(\max L_i, \min R_i)$. То е непразно точно когато $\max L_i < \min R_i$, т.е. когато $L_i < R_j$ за всеки два интервала i и j от множеството. С други думи, няколко интервала имат обща точка тогава и само тогава, когато всеки два от тях се пресичат. Значи е достатъчно да знаем отговорите на заявките за всички двойки.

Колко заявки са нужни. В общия случай по-малко от $N(N-1)/2$ заявки не стигат. Да разгледаме тестове, в които всички интервали са с една и съща дължина и нямат общи точки, освен може би една двойка, която се пресича. Заявка за три или повече интервала тогава винаги връща 0, а заявка за един интервал връща една и съща дължина. Полезна информация дават само заявките за двойки, а всяка от тях проверява само една двойка. Докато не сме питали за всички двойки, не можем да различим теста с отговор 1 от теста, в който непитаната двойка се пресича и отговорът е 2.

Авторското решение задава по една заявка за всяка двойка и по една за всеки интервал, т.е. общо $N(N-1)/2 + N = N(N+1)/2$. Заявките за единичните интервали не са задължителни (по-долу има вариант без тях), но дължините на интервалите правят решението по-просто, а N допълнителни заявки са много малко в сравнение с $N(N-1)/2$.

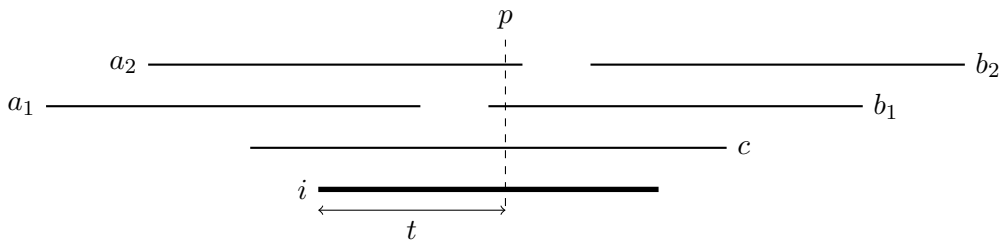
Най-късият интервал. Сортираме интервалите по дължина. При равни дължини ги подреждаме по произволен, но фиксиран начин, например по номер. Всяко множество с обща точка p има най-къс интервал i, т.е. този, който е най-рано в сортирания ред. Всички останали интервали от множеството са след i в реда, пресичат i и съдържат точката p, която лежи в i. Затова за всеки интервал i ще намерим колко най-много от следващите го интервали могат да съдържат една и съща точка от i. Отговорът на задачата е най-добрата от тези стойности.

Защо гледаме точно най-късия интервал? Всеки интервал j след i, който пресича i, е поне толкова дълъг, колкото i, така че има само два случая (вижте фигурата):

- $q(i, j) = d_i$. Тогава j съдържа целия интервал i, а значи и всяка точка от него.

- $0 < q(i,j) < d_i$. Тогава j покрива част от i с дължина $q(i,j)$, която започва от единия край на i.

Ако j можеше да е по-къс от i, отговорът $q(i,j) < d_i$ би могъл да означава и че j е изцяло вътре в i, и нямаше да знаем коя част от i покрива j.



Интервалът i и следващите го интервали, които го пресичат: c съдържа целия i, a_1 и a_2 покриват единия му край (група A), а b_1 и b_2 – другия (група B). Точката p е в интервалите i, c, a_2 и b_1.

Разделяне на две групи. Интервалите от втория вид трябва да разделим на две групи според края на i, който покриват. Не знаем кой край е левият и кой десният, но това не е нужно. Нека m е интервалът от втория вид с най-малко $q(i,m)$ и нека j е друг интервал от втория вид. Ако j покрива същия край като m, той покрива поне толкова от i, колкото и m. Освен това и двата интервала продължават извън i след този край, защото не са по-къси от i. Значи сечението им съдържа частта от i с дължина $q(i,m)$ при този край и още част извън i, така че $q(m,j) > q(i,m)$. Ако j покрива другия край, сечението на m и j е вътре в i, защото m продължава извън i от едната страна, а j – от другата. Тогава $q(m,j) = \max(0,\ q(i,m) + q(i,j) - d_i) < q(i,m)$, защото $q(i,j) < d_i$. Така с проверката $q(m,j) > q(i,m)$ разделяме интервалите на група A (на страната на m, заедно с m) и група B. Отговорите $q(m,j)$ вече са ни известни, така че не задаваме нови заявки.

Избор на точката. Нека p е точка от i на разстояние t от края, който покриват интервалите от A, като $0 < t < d_i$. Интервал j от A съдържа p точно когато $q(i,j) > t$, а интервал j от B – точно когато $q(i,j) > d_i - t$. Ако местим p към другия край на i, интервалите от B, които я съдържат, само стават повече, а интервалите от A не се променят, докато p не стигне до края на някой от тях. Затова е достатъчно да разгледаме две възможности:

- p е точно преди края на някой интервал от A, който покрива част от i с дължина a. Тогава p е в интервалите k от A с $q(i,k) \geq a$ и в интервалите k от B с $q(i,k) > d_i - a$.
- p е точно преди другия край на i. Тогава p е във всички интервали от B и в нито един от A.

И в двата случая към тези интервали добавяме самия i и интервалите, които съдържат целия i.

За фиксирано i разглеждаме $O(N)$ положения на точката и за всяко броим за $O(N)$, така че сложността е $O(N^3)$. Това е достатъчно за тази подзадача.

Вариант само с двойките. Можем да използваме само това кои двойки интервали се пресичат. Интервалът с най-малък десен край има следното свойство: всички интервали, които го пресичат, се пресичат и помежду си, защото съдържат точките точно преди десния му край. Затова многократно намираме сред останалите интервали такъв с това свойство, като проверяваме всички двойки сред интервалите, които го пресичат. Запомняме него заедно с още непремахнатите интервали, които го пресичат, и го премахваме. Ако C е множество с обща точка, в момента, в който премахваме първия интервал от C, останалите интервали от C още не са премахнати и го пресичат. Значи запомненото тогава множество е поне толкова голямо, колкото C. Директната реализация е със сложност $O(N^4)$, което също стига при $N \leq 100$.

Подходи, които дават само част от точките или са грешни.

- За всяка пресичаща се двойка i, j да проверим кои интервали k съдържат цялото ѝ сечение чрез заявки за $\{i, j, k\}$. Това е вярно, защото сечението на множество с обща точка съвпада със сечението на два от интервалите в него: този с най-голям ляв край и този с най-малък десен край. Заявките обаче са от порядъка на N^3 и решението носи само част от точките.
- Да разделяме интервалите на групите A и B чрез допълнителни заявки за по три интервала. Тогава заявките стават повече от $N(N+1)/2$ и се губят точки.
- Да добавяме интервалите един по един, докато сечението остава непразно. Ако в началото добавим интервал, който не е в нито едно оптимално множество, после не можем да го махнем. Затова този подход не гарантира оптимален отговор, дори ако го повторим с много случайни наредби на интервалите.

3 Подзадача 3: $N \leq 5000$, за 30 точки

Заявките са същите, но тук те са около 12,5 милиона и обработка със сложност $O(N^3)$ е твърде бавна. Затова трябва да ускорим избора на точката.

За фиксирано i сортираме интервалите от A и от B по $q(i, \cdot)$. Обхождаме интервалите от A в намаляващ ред на $q(i, \cdot)$. Когато стигнем до интервал с $q(i, k) = a$, всички вече обходени интервали от A и текущият съдържат точката точно преди неговия край. От B я съдържат интервалите с $q(i, k) > d_i - a$. При намаляване на a прагът $d_i - a$ расте, така че тези интервали са последните няколко в сортирания масив на B и броят им само намалява. Затова ги поддържаме с втори указател. Така за фиксирано i работим за $O(N \log N)$ заради сортирането и общата сложност е $O(N^2 \log N)$. Паметта е $O(N^2)$, защото пазим отговорите на заявките за всички двойки.

Вариантът само с двойките също може да се ускори до $O(N^2)$. Графът, в който свързваме всеки два пресичащи се интервала, е хордален, а за намиране на най-голяма клика в хордален граф има известни бързи алгоритми, например Maximum Cardinality Search.

Детайл по реализацията. Следващ интервал със същата дължина като i и с $q(i, j) = d_i$ съвпада с i и попада при интервалите, които съдържат целия i. Затова равните дължини не изискват отделна обработка.