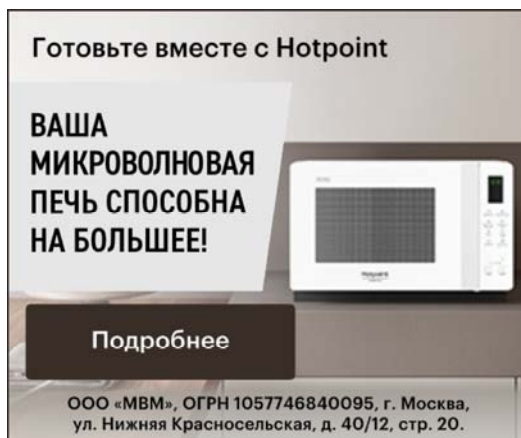


Максимальные интервалы пересечения



Предположим, что у вас есть $k \leq 10^5$ интервалов $[a_i, b_i] \in [1, 10^{18}]$ (некоторые из них могут перекрываться), и вам нужно выбрать набор интервалов, взаимно не пересекающихся, чтобы их объединение было максимальным. Не максимальное количество непересекающихся интервалов, но объединение должно охватывать большинство.

Невозможно попробовать все возможные подмножества 2^k . Жадный подход к упорядочению по a_i (алгоритм интервального покрытия) и упорядочение по b_i (максимальное число алгоритмов непересекающихся интервалов) не сработали. Невозможно выяснить, существует ли динамическое программное решение. Учитывая размер ввода, я думаю, что решение должно быть $O(k \log k)$ или $O(k)$.

Примеры 1. $[1,4], [3,5], [5,9], [7, 18]$ Sol $[3,5]$ и $[7,18]$

2. $[1,2], [2,6], [3,4], [5,7]$ Sol $[1,2]$ и $[3,4]$ и $[5,7]$

3. $[2,30], [25,39], [30,40]$ Sol $[2,30]$

↑ +1 ↓

[algorithm \(/questions/tagged/algorithm\)](/questions/tagged/algorithm)

[intervals \(/questions/tagged/intervals\)](/questions/tagged/intervals)

[greedy \(/questions/tagged/greedy\)](/questions/tagged/greedy)

Moro Silverio (<https://stackoverflow.com/users/7392144/moro-silverio>) 12 мар. '17 в 19:31

источник (<https://stackoverflow.com/questions/42752079/max-coverage-disjoint-intervals>)

поделиться (</questions/13380260/max-coverage-disjoint-intervals>)

2 ответа

Задача может быть решена в $O(k \log(k))$.

Сначала отсортируйте интервалы по их верхним границам (b_i s). Пусть $I(1), I(2), \dots, I(k)$ - список отсортированных интервалов. То есть,

$$b_1 \leq b_2 \leq \dots \leq b_k$$

Обозначим через $w(i)$ длину интервала $I(i)$. То есть,

$$w(i) = b_i - a_i$$

Обозначим через $f(i)$ общую длину оптимального решения среди тех, чей последний интервал $I(i)$. То есть решение, соответствующее $f(i)$ является множеством, которое:

1. содержит интервал $I(i)$
2. не содержит интервала, верхняя граница которого выше b_i
3. имеет максимальное покрытие среди множеств (неперекрывающихся) интервалов, удовлетворяющих 1 + 2

Теперь мы собираемся вычислить $f(1), f(2), \dots, f(k)$ и вернуть максимальное значение из них всех. Ясно, что оптимальное решение соответствует одному из $f(i)$ и поэтому максимальное $f(i)$ является оптимальным решением.



Для вычисления каждого $f(i)$ мы используем динамическое программирование. Мы делаем это, полагаясь на следующее рекуррентное соотношение:

$$f(i) = w(i) + \max\{f(j) \mid b_j < a_i\}$$

Я продемонстрирую вычисления с помощью вашего первого примера ввода:

```
I(1)=[1, 4], w(1)=3
I(2)=[3, 5], w(2)=2
I(3)=[5, 9], w(3)=4
I(4)=[7, 18], w(4)=11
```

Мы вычисляем $f(i)$ для $i=1, 2, 3, 4$:

```
f(1) = w(1) + max{None} = 3
      f(1) intervals: {I(1)}

f(2) = w(2) + max{None} = 2
      f(2) intervals: {I(2)}

f(3) = w(3) + max{f(1)} = 4 + 1 = 5
      f(3) intervals = {I(1), I(3)}

f(4) = w(4) + max{f(1), f(2)} = 11 + f(1) = 11 + 3 = 14
      f(4) intervals = {I(1), I(4)}
```

Максимум $f(i)$ есть $f(4)$ что соответствует множеству интервалов $\{I(1), I(4)\}$, оптимальному решению.

↑ +2 ↓

источник (<https://stackoverflow.com/questions/42752079/max-coverage-disjoint-intervals/42753210#42753210>)

поделиться (</questions/13380260/max-coverage-disjoint-intervals/22124577#22124577>)

Кажется, существует решение $O(k \cdot \log(k))$. Это может быть достигнуто с помощью структуры данных дерева сегментов.

Мы можем сначала заполнить некоторый массив **endPos** **конечных концов** сегмента, отсортировать его. Запомните для каждого из сегментов соответствующий индекс **endPos**. Для этого пусть **endPosIdx** будет таким массивом, что **endPosIdx_j** будет хранить индекс в **endPos**, где заканчивается сегмент **j**-th.

Затем мы введем дерево сегментов. Он обработает следующие запросы:

1. **getMax(i)** - получить максимальное значение в диапазоне $[0, i]$.
2. **update(i, value)** - обновить максимум в позиции **i**-th со **value**.

i is index в массиве **endPos**. Вызов **getMax(i)** мы просим, какую максимальную оболочку мы можем достичь, если не сегмент заканчивается после **endPos_i**. Вызывая **обновление(i, value)**, мы говорим, что теперь существует покрытие с **значением** длины, заканчивающимся на **endPos_i**.

Сортируйте все сегменты в порядке возрастания по их исходной позиции **a_j**. Обработайте их в таком порядке. Суть заключается в том, чтобы найти самую большую оболочку, если мы обязательно возьмем текущий сегмент в результирующем наборе. Текущая оболочка будет равна сумме длины текущего сегмента и максимального покрытия сегментов, заканчивающихся **до** тока. Пусть **j** - индекс текущего сегмента (они сортируются по start pos). Пусть **i** тогда является таким максимальным индексом, что **endPos_i ≤ a_j** (**i** может быть найден из **j** бинарным поиском). Тогда мы можем найти

Открой свой собственный бизнес



Реклама

Реклама Более 200 барбершопов
открыто, 0 закрыто. Каждый...
OldBoy Barbershop

Подробнее

обложка_j = длина_j + getMax(i)

Затем мы должны обновить обновление дерева сегментов (**endPosIdx_j**, **обложка_j**) и перейти к следующему сегменту.

После обработки всех сегментов решение можно найти, вызвав **getMax(размер(endPos))**.

↑ 0 ↓

Ivan Gritsenko (<https://stackoverflow.com/users/1770952/ivan-gritsenko>) 12 мар. '17 в 20:54

источник (<https://stackoverflow.com/questions/42752079/max-coverage-disjoint-intervals/42753044#42753044>)

поделиться (</questions/13380260/max-coverage-disjoint-intervals/22124576#22124576>)

Связанные вопросы

1

Алгоритм для решения максимальных классов, посещаемых студентом
(</questions/17549517/algorithm-to-solve-maximum-classes-attended-by-student>)

Похожие вопросы

- 4 Максимальная сумма интервалов неперекрывающихся интервалов в списке интервалов (/questions/1497328/maximum-sum-of-the-range-non-overlapping-intervals-in-a-list-of-intervals)
- 3 Как быстро вычислить порядок локализации всех пересечений набора множеств (/questions/1643130/how-to-compute-fast-the-containment-order-of-all-intersections-of-a-collection-of-sets)
- 1 Найдите охват набора интервалов (/questions/10159176/find-coverage-of-a-set-of-intervals)
- 1 Застревание в задаче алгоритма графа (/questions/9195898/stuck-on-graph-algorithm-task)
- 1 Взаимное перекрытие подмножества (/questions/11585537/mutually-overlapping-subset-of-activities)
- 0 Внедрение непересекающихся интервальных множеств интервала с TreeSet (/questions/16199800/implementing-disjoint-interval-sets-of-interval-with-treeset)
- 0 Найти максимальную глубину набора интервалов (/questions/14452947/find-the-maximum-depth-of-a-set-of-intervals)
- 1 Жадный алгоритм поиска минимального набора интервалов, перекрывающих все остальные интервалы (/questions/10968539/greedy-algorithm-for-finding-min-set-of-intervals-that-overlap-all-other-intervals)
- 5 Две непересекающиеся интервалы минимальной суммы (/questions/14906260/two-disjoint-intervals-minimum-sum)



IT-курсы для детей 7-14 лет

Обучаем современным IT-навыкам. Растим новые таланты. Ребенок будет в восторге!.

Посмотрите другие вопросы по меткам [algorithm \(/questions/tagged/algorithm\)](/questions/tagged/algorithm)

[intervals \(/questions/tagged/intervals\)](/questions/tagged/intervals) [greedy \(/questions/tagged/greedy\)](/questions/tagged/greedy) или

Задайте вопрос (/want-register)

licensed under cc by-
sa 3.0
(<https://creativecommons.org/licenses/by-sa/3.0/>) with attribution.



Qaru Site