XVII всероссийская командная олимпиада школьников по программированию

11 декабря 2016 года

Задача A
Золотые слитки

Авторы задачи:
Андрей Станкевич,
Олег Христенко
Разработка задачи:
Николай Будин

Решение задачи

- Проверим, можно ли разделить добычу без распила.
- Одному разбойнику достанется один слиток, другому – два.
- Иначе, всегда существует способ с одним распилом.
- Отложим последовательно отрезки длин x_1, x_2 и x_3.
- Отметим точку $(x_1+x_2+x_3) / 2$.
- Распилим слиток, соответствующий отрезку, который поделила эта точка.

Задача B
Автобус

Автор задачи:
Антон Гардер
Разработка задачи:
Михаил Путилин

Постановка задачи

- В автобусе m сидений, рядом с каждым сиденьем можно стоять.
- Когда пассажир входит, он садится на сиденье с наименьшим номером. Если они заняты, то встаёт на стоячее место с наименьшим номером.
- k пассажиров. Про каждого известно, когда он входит и выходит.
- Антон входит первым и может сесть на любое место.
- Нужно выбрать место так, чтобы рядом с Антоном стояли как можно меньше.

Решение

- От чего зависит, будет ли некоторый пассажир сидеть или стоять?
- Только от количества свободных сидячих и стоячих мест на момент его входа
- Поэтому выбор места Антона не влияет на это

Рассмотрим людей, которые стоят

- Человек стоит, если на момент его входа все сиденья заняты
- Место, куда встанет пассажир, не зависит от того, куда сядет Антон
- Можно посадить Антона на любое место, промоделировать процесс, затем выбрать лучшее место

Моделирование

- Отсортируем события: “человек вошёл”, “человек вышел”
- Поддерживаем *set* свободных мест
- Когда пассажир входит, выбираем ему место из множества свободных, удаляем его из множества
- Когда пассажир выходит, добавляем место в *set*
- Время работы: $\boldsymbol{O(k (\log k + \log m))}$
  - Сортировка: $O(k \log k)$
  - Обработка событий: $O(k \log m)$

Задача C
Горные лыжи

Автор задачи:
Михаил Мирзаянов
Разработка задачи:
Станислав Наумов

Постановка задачи

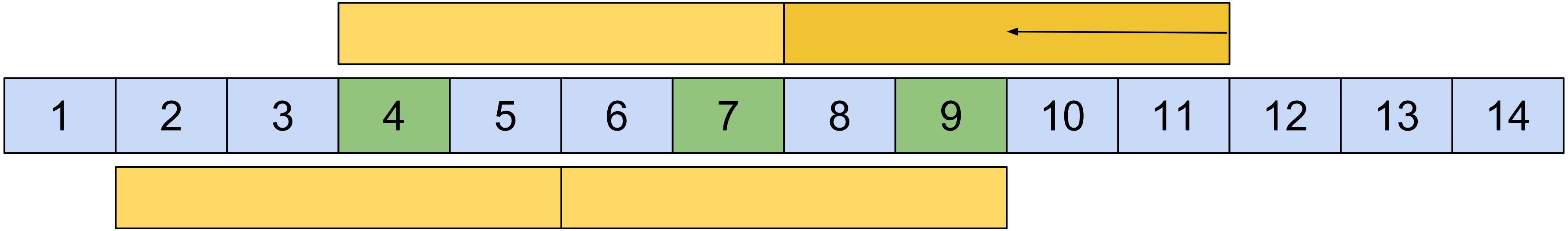
- Задана последовательность *d[i]* - дни, когда Таня точно была на курорте
- Найти максимальное количество зимних дней, в которые девочка могла не быть на горнолыжном курорте.

Упростим задачу

- Пусть нельзя возвращаться после конца зимы

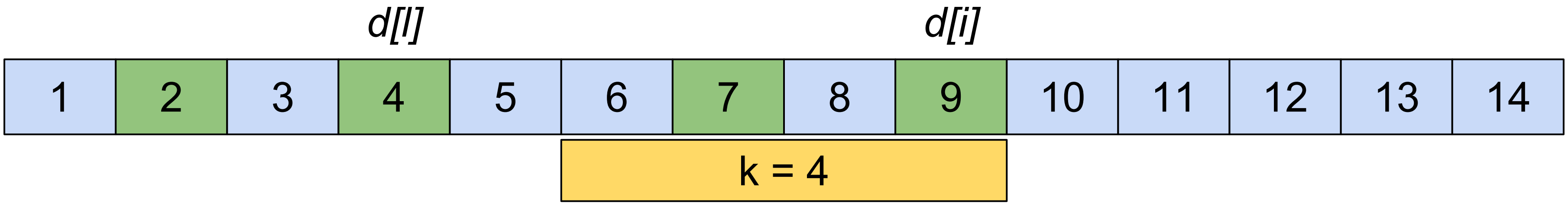
Решение упрощённой задачи

- Отсортируем последовательность дней *d*
- Пусть *dp[i]* - это максимальное количество зимних дней в городе, если Таня вернулась в день *d[i]*
- Тогда нет смысла заканчивать поездку не в какой-то день из *d[i]*
- Если поездка закончилась не в *d[i]*, её можно сдвинуть влево.



Решение упрощённой задачи

- Рассмотрим наибольшее l такое, что $d[l]$ не входит в последний курорт ($d[l] <= d[i] - k$).
- l можно найти бинарным поиском или двумя указателями.
- Если такого l не существует: $dp[i] = max(d[i] - k, 0)$
- Иначе: $dp[i] = dp[l] + d[i] - d[l] - k$



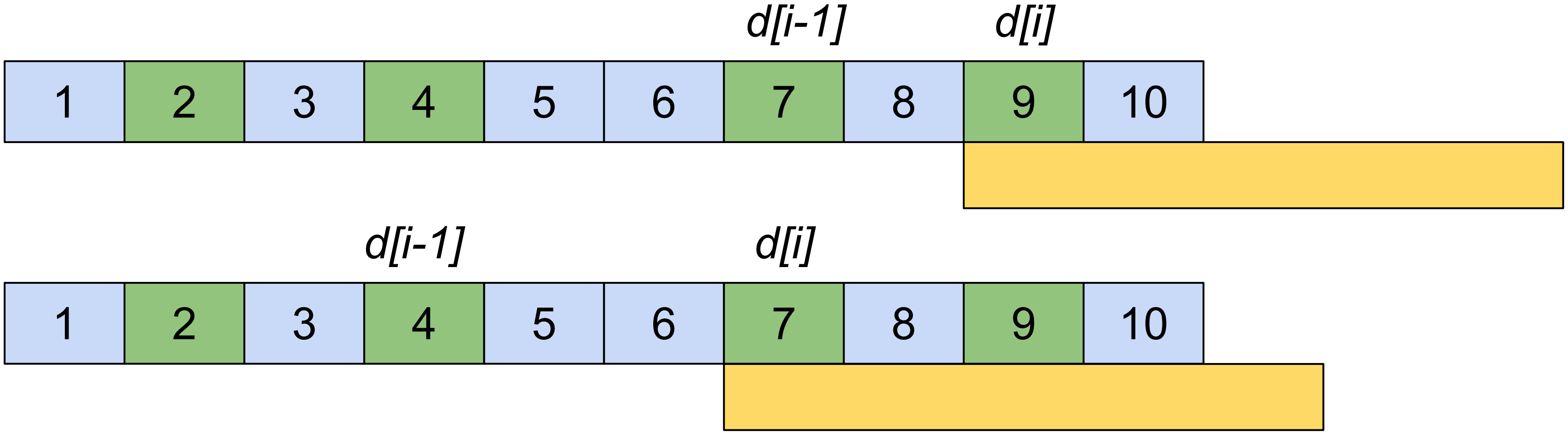
Вернёмся к исходной задаче

- Заметим, что только в одну последнюю поездку Таня могла вернуться весной
- Она началась в один из *d[i]*, иначе её можно сдвинуть вправо

Вернёмся к исходной задаче

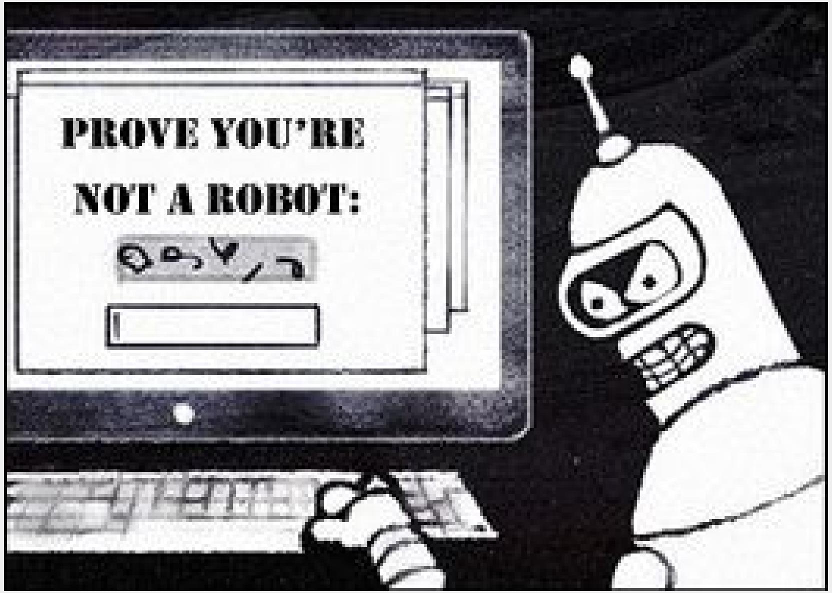
- Переберём этот день. Ответ для дня *d[i]*:
  - *dp[i - 1] + max(n + 1 - (d[i] + k), 0)*
  - Только если эта поездка покрывает все дни от *d[i]* до *d[m]*
- Либо *dp[m]*, если такой поездки не было.
- Ответ - максимум из этих вариантов

Вернёмся к исходной задаче



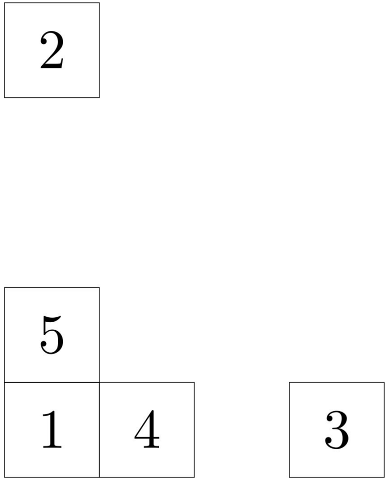
Задача D
IQ тест для роботов

Автор задачи:
Елена Андреева
Разработка задачи:
Николай Будин



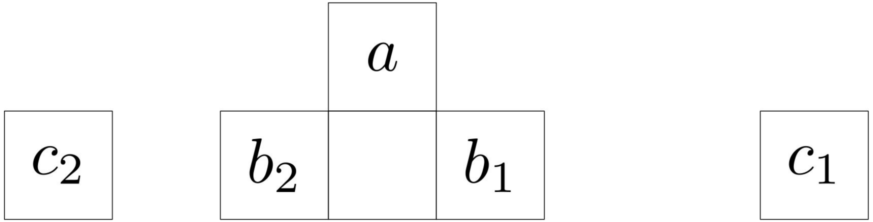
Решение задачи

- Предподсчитаем для каждой клетки ближайшую отличную по цвету клетку в каждом из 4 направлений
- Одна из клеток в ответе всегда будет соседней с клеткой запроса



Решение задачи

- Переберем клетку ответа, соседнюю с клеткой запроса
- Переберем направление, в котором находится вторая клетка ответа
- В качестве второй клетки возьмем либо соседнюю в выбранном направлении, либо ближайшую другого цвета



Задача E
Расшифровка ДНК

Авторы задачи:
Григорий Резников, Станислав Наумов
Разработка задачи:
Николай Будин

Жадное разбиение

- G - префиксное множество
- Разбивать строку $s = g_1 g_2 \ldots g_k$ можно жадно
- Найдем строку из G, которая является префиксом s
- Не более одной такой строки

Решение задачи

- Эмулируем этот процесс для всех строк из *D*
- Поддерживаем бор на строках из *G*
- Для всех строк из *D* поддерживаем самую глубокую вершину в боре, до которой мы дошли

Запрос «? s»

- Смотрим на все символы строки *s*, спускаемся по бору
- Если достигли вершины, соответствующей концу g_i, переходим в корень
- Нет перехода — запоминаем, что в этой вершине есть остаток строки *s*
- Если строка закончилась — выводим в ответ

Запрос «+ s»

- Добавляем *s* в бор
- Когда добавляем новый переход из вершины *v*, спускаем, если нужно, какие-то строки, запомненные в *v*
- Со всеми строками, которые оказались в конечной вершине, сделаем то же самое, что при запросе «? s»
- Выводим все строки, которые полностью разбились

Решение задачи

- Построение бора работает за $O(G)$, где G - суммарная длина всех строк из множества G
- Каждая строка d_i из D передвигается по бору не более $|d_i|$ раз
- Итого все работает за $O(S)$, где S — суммарная длина всех строк

Задача F
Преобразование таблицы

Авторы задачи:
Андрей Лопатин,
Андрей Станкевич
Разработка задачи:
Дмитрий Филиппов

Постановка задачи

- Есть таблица $w \times h$, где в ячейке $[i, j]$ записано $i \cdot w + j$
- Поступают запросы - поменять местами два столбца/строки/клетки таблицы
- Запросов много, таблица большая
- После выполнения всех запросов вывести контрольную сумму итоговой таблицы

Решение задачи

- Будем хранить перестановки столбцов и строк - *pc* и *pr* соответственно
- *pc[i]* = *x* означает, что в *i*-м столбце сейчас на самом деле стоит столбец номер *x* изначальной таблицы
- Аналогично для *pr*

Как делать запросы?

- *c x y - swap(pc[x], pc[y])*
- *r x y - swap(pr[x], pr[y])*
- В клетке (a, b) находится *matrix[pr[a]][pc[b]]*
- Аналогично в (c, d) находится *matrix[pr[a]][pc[b]]*
- *f a b c d - swap(matrix[pr[a]][pc[b]], matrix[pr[c]][pc[d]])*

Решение

- После выполнения всех запросов, применим перестановки pc и pr к таблице и посчитаем контрольную сумму
- Каждый запрос выполняется за $O(1)$, итоговая асимптотика: $O(wh + q)$

Задача G
Архивы джедаев

Автор задачи:
Борис Минаев
Разработка задачи:
Михаил Путилин

Постановка задачи

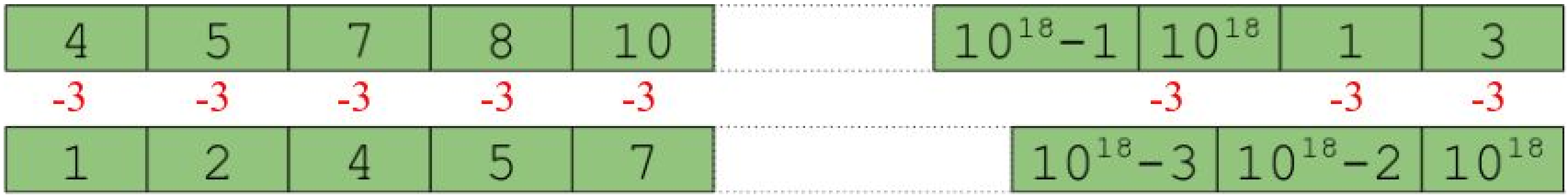
- Был массив длины 10^{18}, индексация с единицы, $a[i] = i$
- Выкинули m элементов, остальные сдвинули
  - $0 \le m \le 500$
- После этого массив циклически сдвинули
- Дано m и число x, $1 \le x \le 10^{18}$
- Нужно найти в массиве число x (или сказать, что его нет)
- Можно сделать *10* запросов “посмотреть на i-й элемент”.

Что если не было циклического сдвига?

- Изначально число *x* стояло на позиции *x*
- При удалении элементов оно сместилось влево не более чем на *m*, либо было удалено
- Бинарный поиск с границами *x-m* и *x*
- $m \le 500$, *9* запросов

Полное решение

- Посмотрим на элемент *a[1]*. Пусть это *y*.
- Мысленно вычтем из всех элементов *y-1*
- Если получилось число < *1*, добавим к нему 10^{18}
- Теперь у нас предыдущий случай
- *1* запрос к *a[1]* плюс *9* на бинпоиск



Задача H
Музей

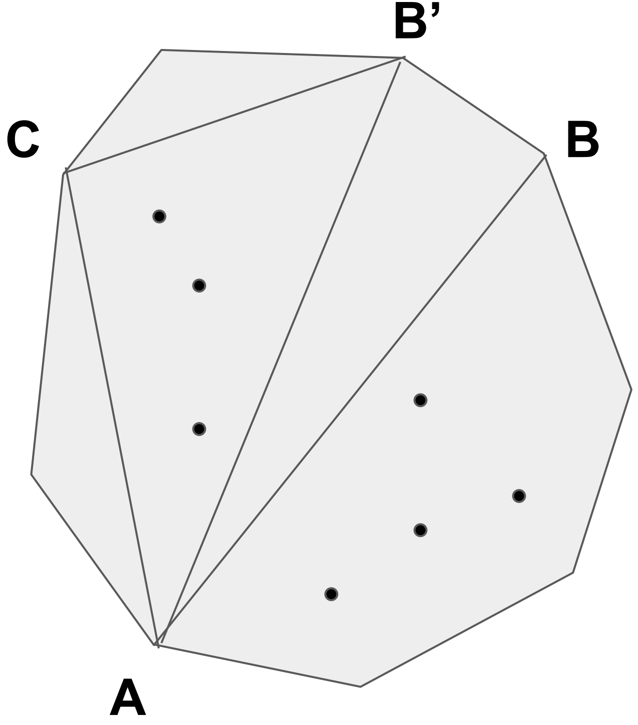
Автор задачи:
Андрей Станкевич
Разработка задачи:
Антон Гардер,
Михаил Пядеркин

Постановка задачи

- Есть зал – выпуклый многоугольник с N вершинами
- Внутри него расположено M точек
- Нужно выбрать один или два не пересекающихся треугольника с вершинами в углах зала, чтобы каждая из M точек содержалась внутри какого-то треугольника, а суммарная их площадь была минимальна

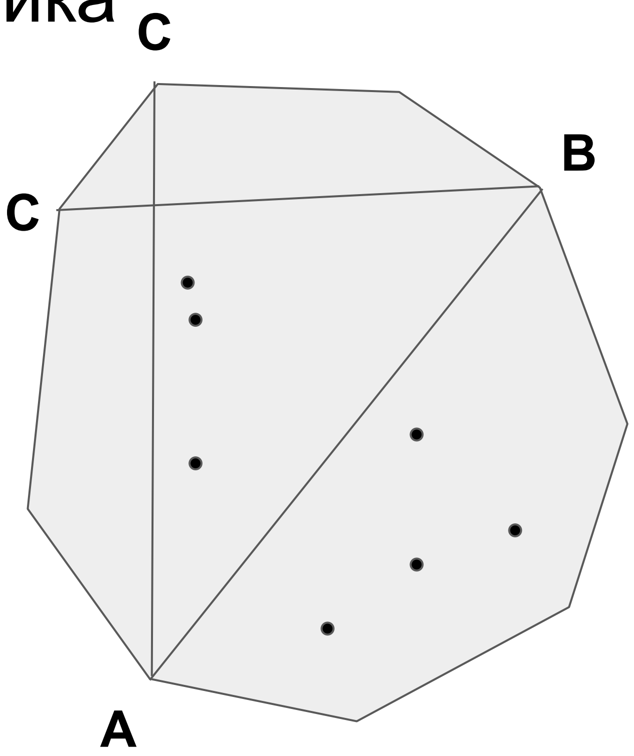
Рассмотрим диагонали

- Любая диагональ разбивает множество точек на две части
- Обозначим за opt[A][B] минимальный по площади треугольник с вершиной в точке A, содержащий все точки слева от $\overrightarrow{AB}$
- Ответ – минимум по всем opt[A][B] + opt[B][A]



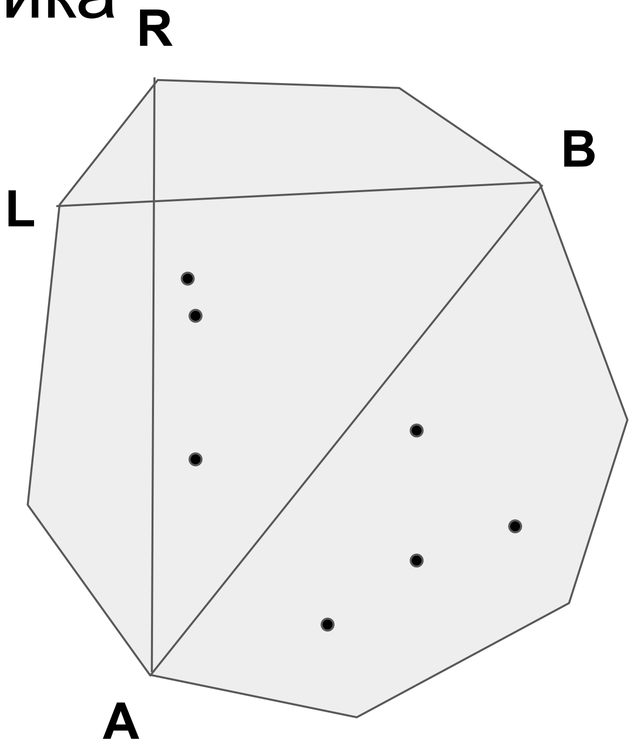
Пусть AB – сторона треугольника

- Как найти третью вершину?
- Какая-то вершина C подходит, если слева от $\overrightarrow{AC}$ нет точек и справа от $\overrightarrow{BC}$ нет точек
- Оба этих условия дают два отрезка подходящих вершин



Пусть AB – сторона треугольника

- Обозначим за L и R концы пересечения этих отрезков
- При движении точки C от L к R расстояние от C до AB сначала неубывает, а затем невозрастает (т. к. зал выпуклый), а значит минимум расстояния достигается в концах отрезка



Иначе

- Рассмотрим случай, когда AB не является стороной треугольника
- Но все такие треугольники мы уже рассматривали, когда находили opt[A][B'], где AB' – предыдущая рассмотренная диагональ
- Проблема может быть лишь в том случае, когда между AB и AB' есть точки

Иначе

- Чтобы проверить этот случай, отсортируем все сувениры по полярному углу относительно A и будем поддерживать указатель на последний сувенир, находящийся левее текущей диагонали
- Тогда, переходя к следующей диагонали, просто проверим, изменится ли этот сувенир

Задача I
Обычный мальчик

Автор задачи:
Денис Кириенко
Разработка задачи:
Станислав Наумов

Постановка задачи

- Дано число N
- Найти любое x на отрезке [N, 1.01 N] с не менее 100 делителями

Решение задачи

- Если у числа h не менее 100 делителей, то для любого натурального t, h×t имеет не менее 100 делителей
- Выберем минимальное число со 100 делителями: 45360
- Для N меньше 4536000 решим перебором
- O(N / 100 * sqrt(N))
- Для N не меньше 4536000 ответ
(N div 45360 + 1) × 45360

Задача J
Полиглоты-интроверты

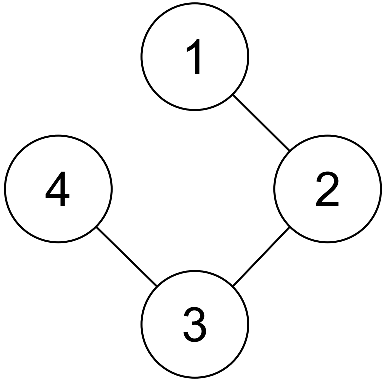
Автор задачи:
Глеб Евстропов
Разработка задачи:
Артем Васильев

Постановка задачи

- Есть n людей, которые знают какие-то из k языков
- Можно сказать фразу на каком-то языке, и ее услышат все, кто знает этот язык
- Как передать фразу от i-го человека к j-му, чтобы ее услышало как можно меньше людей?

Граф языков

- Вместо ответов для пар людей ищем ответы для пар языков
- Граф на языках: ребро из A в B, если есть человек, который знает как A, так и B

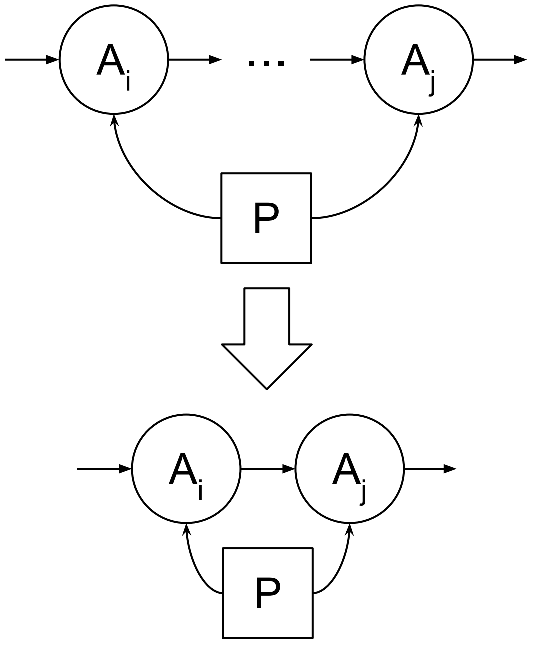


Решение за $O(n \cdot k \cdot 2^k)$

- Переберем множество языков, использующихся в ответе
- Если это множество языков связно, то можно обновить ответ для всех пар языков в множестве

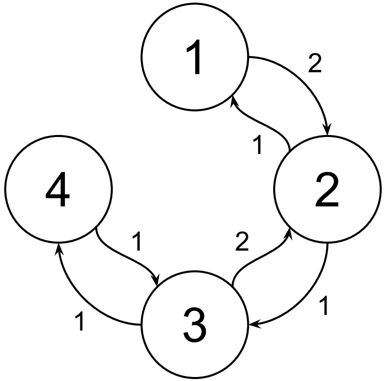
Сокращение цепочек языков

- Пусть для цепочки языков есть человек, которые знает два не соседних языка
- Можно сократить цепочку, улучшив ответ
- В оптимальном ответе пересечение по людям только в соседних языка



Взвешенный граф языков

- Вес ребра A → B равен количеству людей, знающих B, но не знающих A
- Оптимальная цепочка из A в B - кратчайший путь A ⇝ B плюс все люди, знающие A
- Все пары кратчайших путей - за $O(k^3)$ алгоритмом Флойда

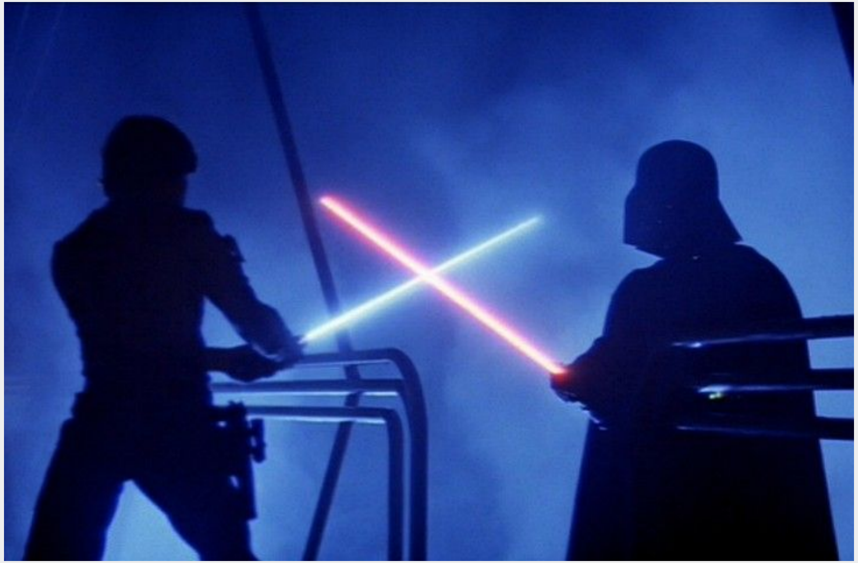


Восстановление ответа

- За $O(n^2k^2)$: для всех пар людей перебрать все пары начальных и конечных языков
- Быстрее: предподсчитать $P_{i,\ j}$ — оптимальная цепочка, начинающаяся в языке, который знает i-й человек, и заканчивающаяся в j-м языке
- Итого: $O(k^3 + nk^2 + n^2k) = O(max(n, k)^3)$

Задача K «Исключающее или» наносит ответный удар

Автор задачи:
Михаил Пядеркин
Разработка задачи:
Михаил Пядеркин



Первое решение

- *a xor a=0* и 0 делится на n
- Это означает, что ответ никогда не превосходит $a < 2^{60}$
- Будем генерировать ответ по битам, начиная со старших
- Выгодно ставить 0 в очередной бит, если это возможно
- Задача сводится к определению того, существует ли искомое *b* с некоторым фиксированным началом

Первое решение

a	0	1	1	0	1	...
b	1	0	?	?	?	...
a xor b	1	1	?	?	?	...

- Числа вида *a xor b*, где *b* имеет фиксированное начало, образуют некоторый непрерывный отрезок
- Необходимо проверить, что отрезок содержит число, кратное *n*
- $l >= r - r \bmod n$

Второе решение

- Заметим, что $|(a \text{ xor } b) - a|$ никогда не будет большим
- Легко доказать, что эта величина не превосходит $2n$, а на самом деле - меньше n
- Так как $a \text{ xor } b$ должно делиться на n, то это значит, что
  - либо $a \text{ xor } b = a - a \bmod n$
  - либо $a \text{ xor } b = a - a \bmod n + n$
- Из двух вариантов нужно выбрать минимальное b

Спасибо за внимание!