Условие задачи

- Задан поезд с m посадочными местами, который едет со станции 1 до станции n
- Заданы k купленных билетов на него
  - Купленный билет задается станцией начала, конца и занятым местом
- Заданы q запросов: начальная и конечная станции
- Требуется найти какое минимальное число билетов нужно купить, чтобы добраться от начальной до конечной станции

Решение

- Утверждение: если мы хотим купить билет на станции v, то надо покупать билет на то место, которое будет свободным как можно дольше
  - Всегда можно сойти раньше, если надо
- $f[v]$ — максимальный номер станции, до которой можно добраться, купив **один** билет на станции v
- Если мы построим f, можем отвечать на запрос (s, t) за $O(n)$:
  - Пусть v — станция, на которой мы находимся, изначально $v = s$
  - Если $f[v] \geq t$, то мы можем за один билет доехать до конечной станции
  - Иначе, надо за один билет ехать до станции $f[v]$ и продолжать путь оттуда

Подзадача 1

- Ограничения: $n, m, k \leq 100$
- Ограничения позволяли для всех пар станций понять, можно ли между ними проехать за одну поездку
- После чего применив метод динамического программирования или алгоритм поиска кратчайших путей в графе найти ответ
- Либо построить массив f не самым оптимальным способом и ответить на каждый запрос за линейное время

Подзадача 2

- Ограничения: $n, m, k \leq 200\,000$ и $q = 1$
- В этой подзадаче требуется посчитать функцию f оптимально
- Для каждого места отсортируем все билеты по возрастанию начальной станции
- Сгенерируем все отрезки пути, когда это место свободно
- Создадим события: начало отрезка, конец отрезка
- Методом сканирующей прямой справа налево пройдемся и будем хранить максимальную станцию, до которой можно доехать
  - Для этого можно использовать дерево отрезков, кучу или дерево поиска
- Ответим на единственный запрос за $O(n)$

Подзадача 3

- Ограничения: $n, m, k, q \leq 200\,000$
- В этой подзадаче требуется посчитать функцию f так же, как и в предыдущей подзадаче
- Но отвечать на запрос надо быстрее
- $g[i][v]$ — до какой станции можно добраться от станции v, если купить ровно 2^i билетов
  - $g[0][v] = f[v]$
  - $g[i][v] = g[i-1][g[i-1][v]]$
- После чего за $O(\log n)$ можно ответить на запрос
  - По убыванию i пытаться покупать сразу 2^i билетов
  - Если, купив 2^i билетов, мы еще не добрались до конечной станции, то купить
    - $g[i][v] < t \Rightarrow v := g[i][v]$
  - В самом конце мы окажемся в одном билете от конечной станции