Разбор задачи 8. Магические порталы

Региональный этап Всероссийской олимпиады школьников по информатике
Республика Татарстан. Казань

26 января 2015

Условие задачи

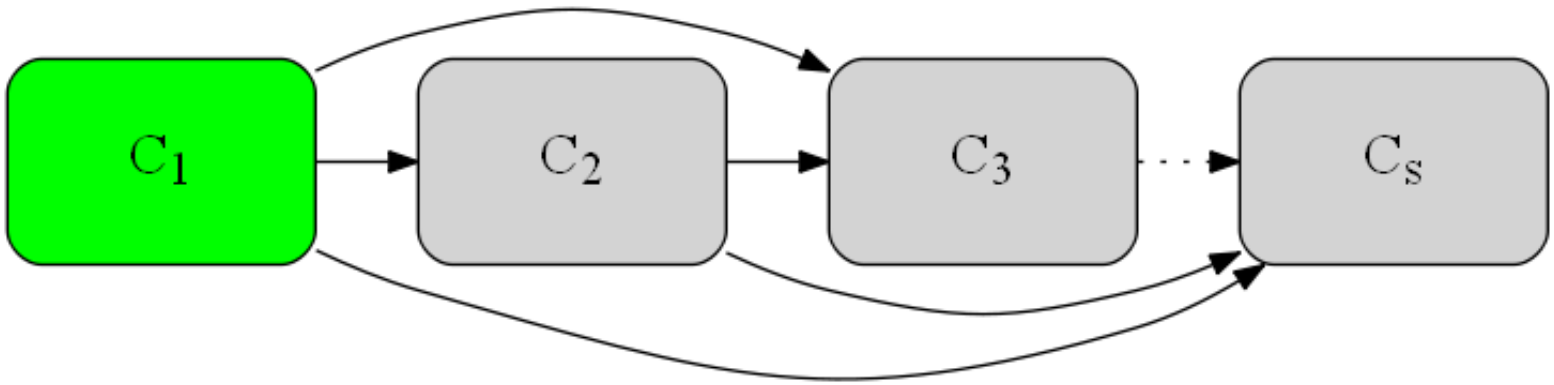
- Задан турнир: ориентированный граф, между любыми двумя вершинами которого есть ровно одно ребро
- Вершина называется совершенной, если из нее достижимы все вершины графа
- Требуется для каждого k посчитать, сколько существует ребер, что если его развернуть, то в графе будет ровно k совершенных вершин

Подзадача 1

- Турнир состоит не более чем из 50 вершин и ответ нужно вывести только для k, больших чем число совершенных вершин в изначальном графе
- Можно перебрать все пары вершин, развернуть ребро между ними, и проверить каждую вершину на совершенность
- Итоговое время работы $O(n^5)$

Немного теории

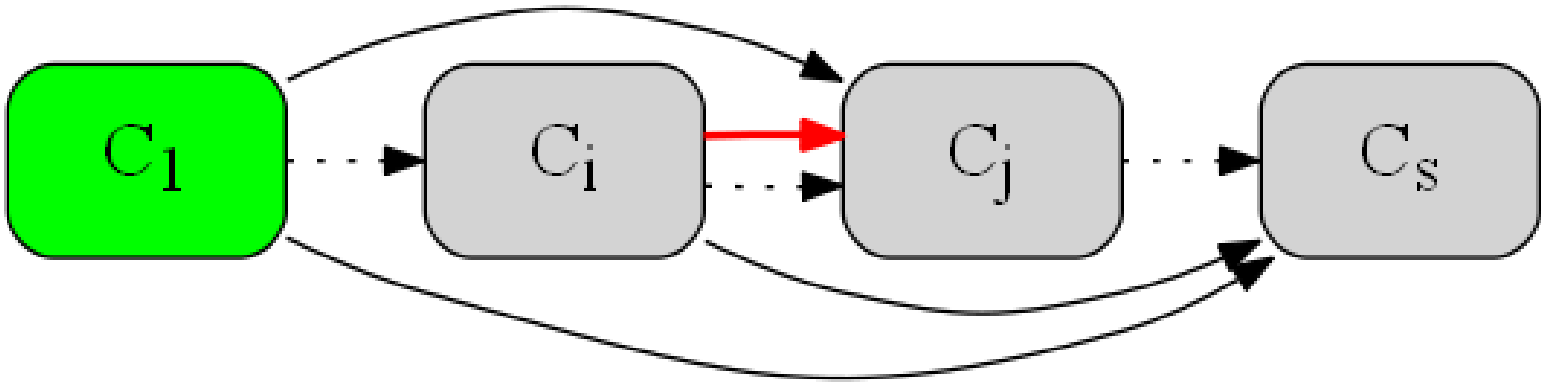
- Верно утверждение, что в любом турнире есть гамильтонов путь (путь, который проходит по всем вершинам графа ровно один раз)
- Если рассмотреть в турнире все компоненты сильной связности и сделать конденсацию, то полученный граф будет выглядить как один путь: $c_1 \to c_2 \to \ldots \to c_s$



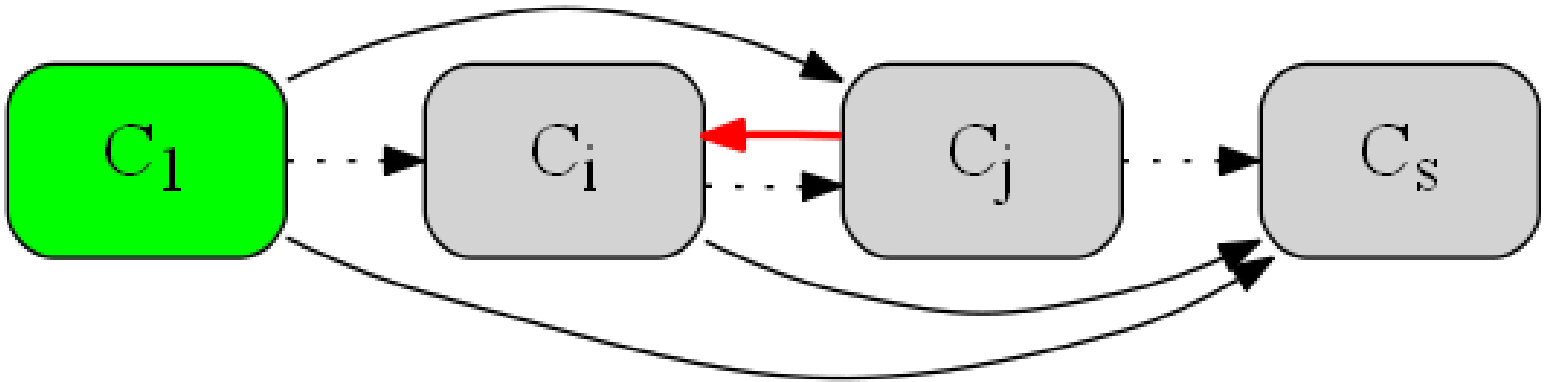
- Вершины, которые лежат в компоненте сильной связности c_1, являются совершенными и только они

Случай первый

- Рассмотрим пару вершин турнира: v и u, лежащие в c_i и c_j, при $1 < i \leqslant j$



- Тогда если развернуть ребро vu множество c_1 не изменится

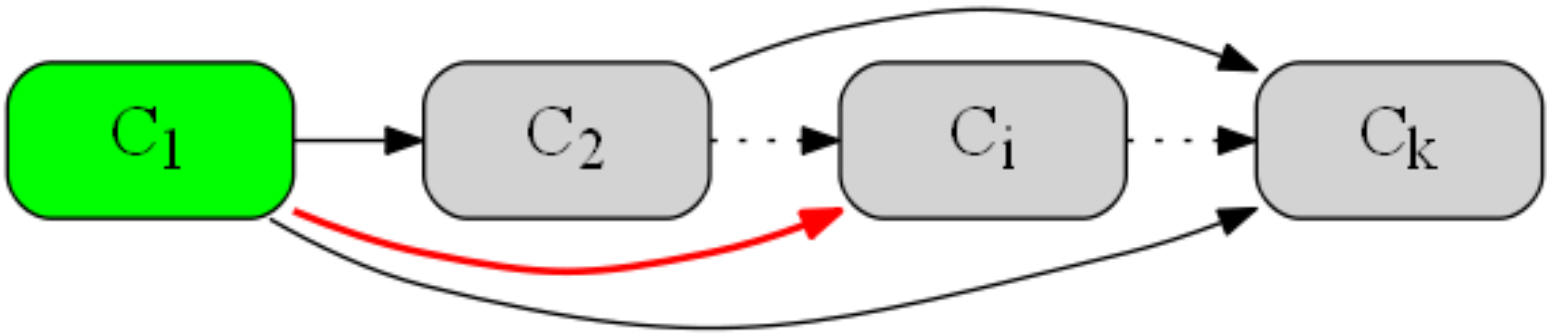


Случай второй

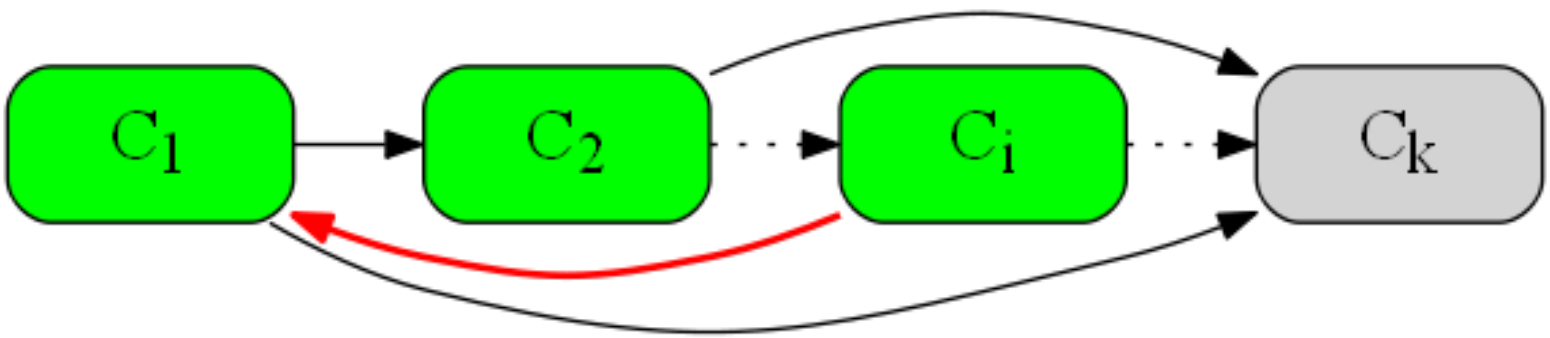
- Если v и u лежат в c_1, то число совершенных вершин может только уменьшиться, что не относится к нашей подзадаче, этот случай мы рассмотрим в Подзадаче 4

Случай третий

- Значит v лежит в c_1, а u лежит в c_i, где $i > 1$

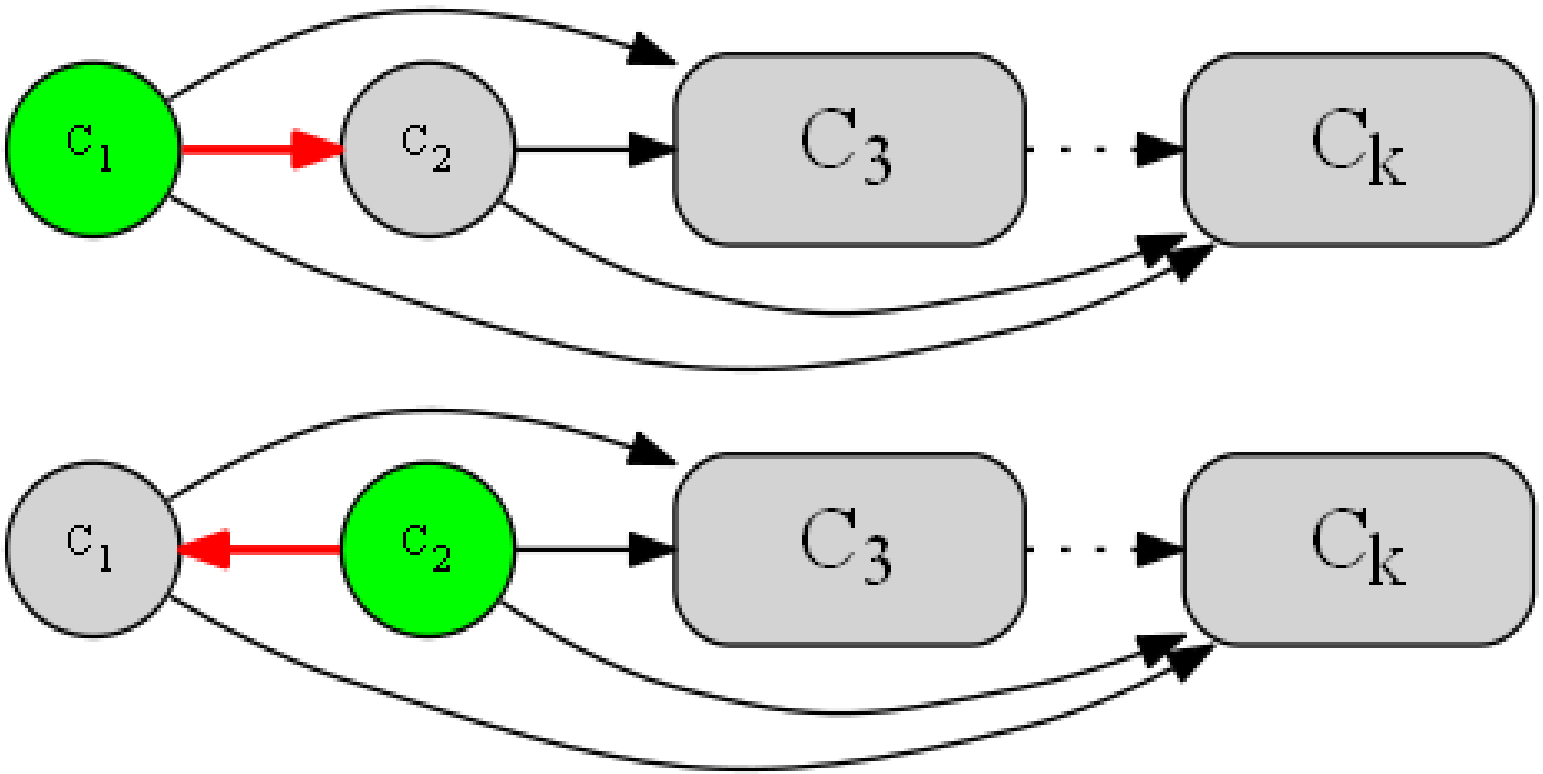


- Развернем ребро vu, теперь из всех компонент сильной связности c_j, где $1 \leqslant j \leqslant i$, можно добраться до компоненты c_1, а значит $ans[c_1 + c_2 + \ldots + c_i]$ увеличить на $c_1 c_i$



Исключение

- Кроме одного случая, когда v перестает быть в c_1: $|c_1| = |c_2| = 1$ и u лежит в c_2, этот случай нужно обработать отдельно, он не меняет число совершенных вершин



Подзадача 2

- Вершин не более 300 и ответ нужно вывести только для k, больших чем число совершенных вершин в изначальном графе
- Задача свелась к нахождению компонент сильной связности и построению конденсации графа
- В этой подзадаче это можно было делать, например, алгоритмом Флойда
- Найти все пары вершин, которые достижимы друг из друга и объединить в одну компоненту
- Время работы: $O(n^3)$

Подзадача 3

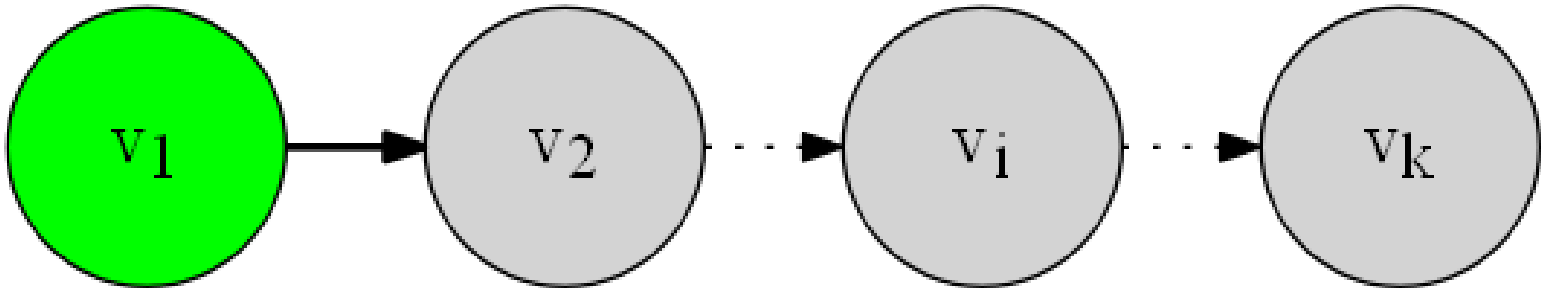
- Вершин не более $2\,000$ и ответ нужно вывести только для k, больших чем число совершенных вершин в изначальном графе
- Применяем все те же идеи, что и для прошлой подзадачи, только строим конденсацию за $O(n^2)$

Подзадача 4

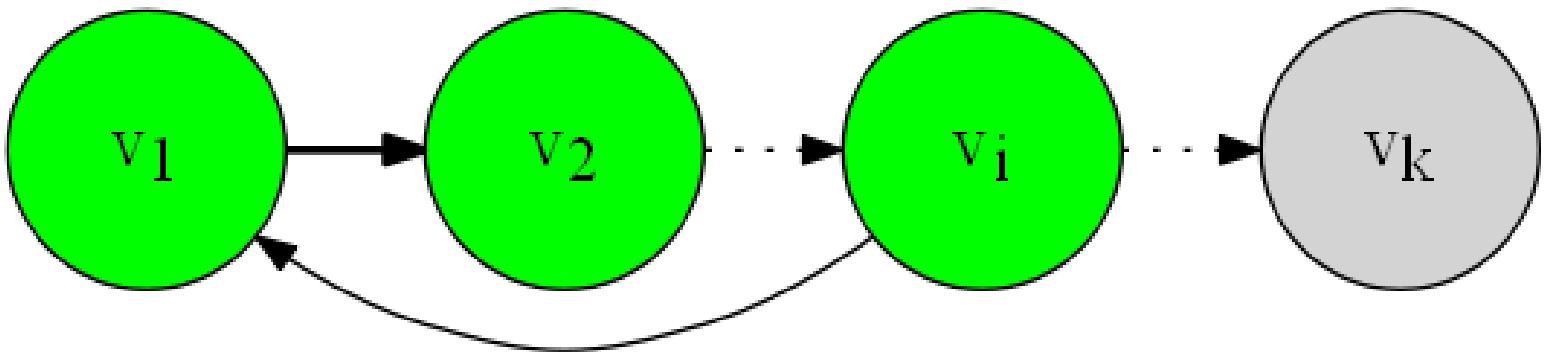
- Вершин не более $2\,000$
- Для k больших чем число совершенных вершин в изначальном турнире нужно применить идею из Подзадачи 3
- Осталось рассмотреть случай, когда количество совершенных вершин уменьшится. Это может произойти только тогда, когда мы разворачиваем ребро vu, а v и u лежат в c_1
- Следующее утверждение верно: в сильносвязном турнире есть гамильтонов цикл
- Найдем гамильтонов цикл в c_1: $v_1, v_2 \dots v_s$
- Если мы развернем ребро не из гамильтонова цикла, то компонента сильной связности не изменится

Подзадача 4

- Развернем ребро $v_s v_1$

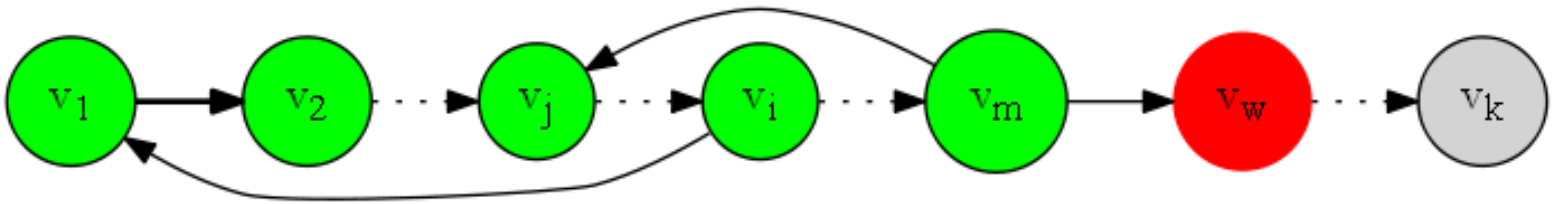


- Заметим, что v_1 осталось совершенным, для этой вершины множество достижимых не поменялось
- Если из v_i достижимо v_1, то v_1 также достижимо из v_{i-1}



Подзадача 4

- Нужно найти минимальное w, при котором из v_w в v_1 нет пути



- Для каждой вершины v_i найдем максимальное $f_i = j$, что $v_j \to v_i$, это значит, что из всех вершин v_k, если $k < j$, можно добраться по ребрам до v_i

Подзадача 4

- Будем добавлять по одной вершине и проверять, максимум f_i по всем добавленным вершинам — это такое число, что из v_{f_i} достижимо v_1, назовем этот максимум M
- Если $M <= i$, то мы нашли $w = M + 1$, потому что из v_{M+1} уже никак не добраться до v_1
- Чтобы перейти к следующему ребру, надо v_1 перекинуть в конец пути, и пересчитать f_i за линейное время
- Итого, время работы: $O(n^2)$