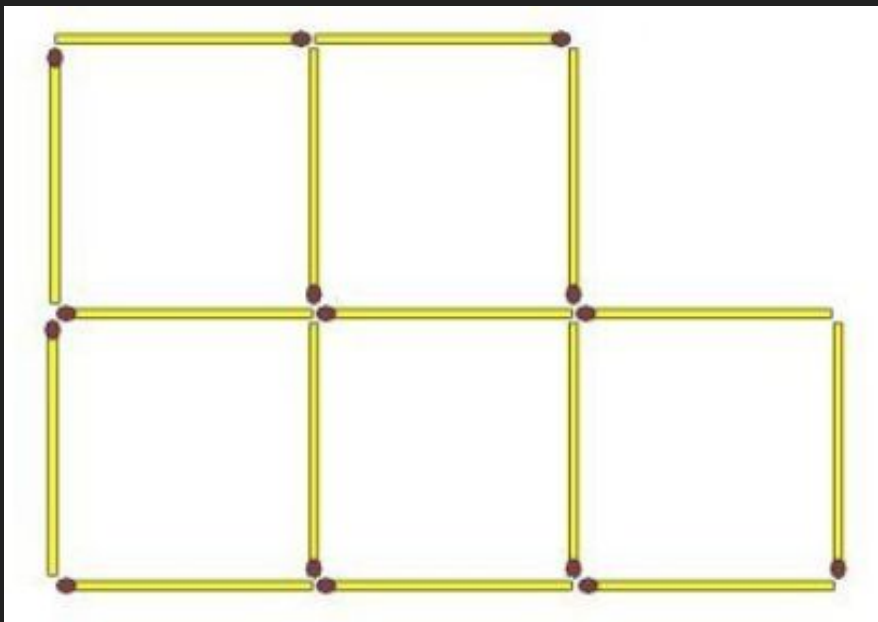


Теория игр

Поиграем?

На столе лежат пятнадцать (15) спичек. Игрок может взять одну, две или три спички. Выигрывает игрок, взявший последнюю спичку.



Поиграем?

Пусть мы играем на позиции первого игрока. Тогда возьмём 3 спички, чтобы сумма на столе делилась на 4 ($k+1$), а потом просто будем добирать до такой суммы и когда-то доберём до нуля.

При этом второй игрок никак не может выиграть (первый всегда имеет возможность сделать так, чтобы сумма двух последних ходов стала равна 4)

Эта игра выигрышная для первого игрока и проигрышная для второго.

Теория комбинаторных игр

Два игрока ведут игру с полной информацией, делая ходы по очереди.

В отличие от “классической” теории игр, где игроки ходят одновременно (дилемма заключённого, например)

Классификация комбинаторных игр

Справедливые -- возможные ходы игроков совпадают (оба могут взять по 1, 2, 3 спички)

Нормальные -- игрок, который не может сделать ход, проигрывает.

Противоположные, ненормальные игры называются “игрой в поддавки”

Случайные -- после хода каждого из игроков производится некоторое случайное действие, изменяющее позицию на игровом столе

Игры на графах

Пара $A = \langle G, u \rangle$, G -- ориентированный граф, u -- одна из вершин (начальная позиция). Каждая вершина представляет некоторое положение игрового стола, каждое ребро -- ход, переводящий игру из одного положения в другое.

Если в G есть ребро uv , а игра $A = \langle G, u \rangle$, $B = \langle G, v \rangle$, то говорят, что из игры A возможен ход в игру B .

Процесс игры: на начальную вершину ставится некая фишка, которую игроки могут по очереди передвигать по рёбрам графа. Игрок, который не может переместить фишку (находится в терминальной вершине), проигрывает

Как же определить, кто выигрывает?

Пока что будем работать с ациклическими графами, то есть игра не может продолжаться бесконечно и нельзя вернуться в положение игрового стола, которое уже встречалось в игре.

Тогда игра A называется выигрышной, если вне зависимости от действий второго игрока первый побеждает. A называется проигрышной, если побеждает второй игрок.

Игра на ациклическом графе

Теорема 1. Все вершины ациклического графа $G = \langle V, E \rangle$ можно разбить на два непересекающихся множества N и P (выигрышных и проигрышных вершин). При этом, если вершина выигрышная, из неё есть ход в проигрышную. А если вершина проигрышная, все ходы из неё ведут в выигрышные вершины.

$$N = \{u \mid \exists (u,v) \in E \text{ т.ч. } v \in P\}$$

$$P = \{u \mid \forall (u,v) \in E \text{ } v \in N\}$$

Игра на ациклическом графе

Доказательство.

Докажем индукций по обратной топологической сортировке графа.

База: пустой граф (или граф на одной вершине) можно разбить таким образом.

Индукция: уже определена классификация для всех вершин, меньших по номеру в топологической сортировке. Тогда если хотя бы одна вершина из текущей ведёт в проигрышную, текущая объявляется выигрышной. Иначе -- проигрышной.

Ретро-анализ

Доказательство теоремы описывает в том числе алгоритм определения, является ли игра выигрышной или проигрышной (пока что только на ациклических графах)

Таким образом за $O(V+E)$ (Стоимость нахождения обратного топсорта + вычисления класса для каждой вершины) можем проанализировать граф игры. Ещё дешевле (на константу) -- при обходе в глубину, возвращаясь из уже рассмотренной вершины, передавать данные о ней “наверх” и таким образом заполнять граф.

Ретро-анализ

Более того, такое разбиение позволяет найти выигрышную стратегию для игроков (всегда, если это возможно, ходить в проигрышные вершины)

Стратегия: отображение s из множество нетерминальных вершин в множество всех вершин графа,

при чём $\square x = s(y)$ должно выполняться $(y, x) \in E$

Игры на графах с циклами

Пусть в графе игры есть циклы. Тогда проигрывающий игрок захочет постоянно ходить по циклу, не приходя в терминальную вершину и не проигрывая. Как же анализировать такой граф и искать для него стратегию?

Примем, что игрок, стартующий в выигрышной вершине, хочет выиграть за минимальное количество ходов, а игрок, стартующий в проигрышной, хочет как можно дольше не проигрывать. Введём классы N_i (вершины, из которых выигрыш достигается за i ходов) и P_i (вершины, из которых проигрыш достигается за i ходов)

Игры на графах с циклами

Рассмотрим граф $G = \langle V, E \rangle$

P_0 -- терминальные вершины графа, проигрыш за 0 ходов

$$N_1 = \{ u \mid \exists v \in P_0 : (u, v) \in E \}$$

$P_2 = \{ u \mid \exists v : (u, v) \in E, v \in N_1 \}$ и так далее

Стоит отметить, что $P_i \subset P_{i+2}$ и $N_i \subset N_{i+2}$.

Тогда класс P -- объединение P_i , N -- объединение N_i

Игры на графах с циклами

Теорема 2.

Пусть P -- множество проигрышных вершин, N -- множество выигрышных.
Тогда $D = V \setminus (P \cup N)$ -- множество ничейных вершин.

Доказательство.

P и N -- аналогично случаю ациклических графов. Покажем, что остальные вершины ничейные, то есть из них нельзя победить, зато можно не проиграть.

Игры на графах с циклами

Пусть фишка находится в некоторой вершине. Если из неё есть ребро в проигрышную вершину -- сходим по ней и считаем исходную выигрышной. Если все рёбра ведут в выигрышные -- идём по любому и проигрываем, считая исходной проигрышную. Пусть из вершины есть ребра в выигрышные и ничейные. Идём в ничейную, чтобы не проигрывать и объявляем исходную тоже ничейной. Применяем те же рассуждения к новой вершине и так до бесконечности.

Как теперь определить, сможем ли мы выиграть?

Используем подобие обхода в глубину. Надо не только определить, выигрышная вершина или проигрышная, но и найти $c(u) == \min \{i, u \in N_i \text{ или } u \in P_i\}$

Для каждой вершины заведём счётчик $z(u)$. Будем хранить в нём количество вершин, до которых можно дойти по ребру из u и которые ещё не определены как $\in N$. Если этот счётчик обнуляется вершина признаётся проигрышной

Алгоритм

Терминальные вершины исходно имеют нулевой счётчик и помещаются в P_0

Используем очередь, в которую помещаются вершины из P_0 .

Рассмотрим очередную вершину u из очереди. Если вершина проигрышная, то все вершины, из которых есть ребро в u , становятся выигрышными, при том, если вершина ещё не проинициализирована, то поместим её в $N_{c(u)+1}$ и $c(v) = c(u) + 1$, добавим в очередь.

Если вершина u выигрышная, то для каждой v , т.ч. $(u,v) \in E$, $z(v) = 0$. Если при этом $z(v) = 0$, то положим v в $P_{c(u)+1}$, $c(v) = c(u)+1$, добавим v в очередь.

Ретро-анализ

Вершины, которые остались не классифицированы после опустошения очереди, объявляются ничейными.

Каждая дуга будет просмотрена 1 раз, значит алгоритм работает за $O(E)$.
Корректность можно доказать индукцией по числу шагов.

Данный алгоритм называют Ретро-анализом

Задача:

Пусть играем в “поддавки”. Как провести ретро-анализ для такой игры?

Пишите в комментарии ;)