**Решаем задачи на графе.** Задачи о поиске максимальной клики, максимального независимого множества и минимального контролирующего множества сводятся друг к другу, поэтому если мы решили одну задачу (за какое-то $O(f(n))$), то и другие задачи решили. Будем решать задачу о поиске минимального вершинного покрытия.

**Асимптотические оптимизации, которые не казались такими сначала.** Задачу о вершинном покрытии можно решать за $O(2^n m)$: для каждой вершины решаем, берем мы ее или нет:

$$t(G, V) = t(G \setminus v, V \cup \{v\}) + t(G \setminus v, V)$$

Заметим, что если мы решили не брать какую-то вершину в ответ, то мы обязаны взять всех ее соседей. Тогда мы можем решать задачу на каком-то меньшем графе, а именно:

$$t(G, V) = t(G \setminus v, V \cup \{v\}) + t(G \setminus \{v \cup N(v)\}, V \cup N(v))$$

Рассмотрим максимальную степень вершины в графе $maxd$. Если, $maxd = 0$, то задача решается за $O(1)$. Тогда пусть $maxd \geq 1$. Таким образом, во втором случае число вершин уменьшается на два:

$$t_n = t_{n-1} + t_{n-2}$$

Получили решение за $O(\phi^n m)$, как оценка на числа Фибоначчи.

Давайте сделаем теперь $maxd \geq 2$ — когда у нас остались ребра, для каждого ребра возьмем одну вершину:

$$t_n = t_{n-1} + t_{n-3}$$

$$x^3 - x^2 - 1 = 0$$

Получили $O(1.47^n m)$.

Если $maxd = 2$, то граф разбивается на пути и циклы, в которых поиск вершинного покрытия тривиален. Аналогично получаем решение за $O(1.38^n m)$.

**Перебор в антагонистических играх.** Нам дано полное двоичное дерево четной глубины $2n$. Фишка стоит в корне дерева. Каждый игрок на своем ходу перемещает фишку налево или направо. В каждом листе написано «0» или «1», причем мы не знаем значения в листах заранее, а можем только спрашивать у оракула (который не играет против нас, то есть неадаптивный).

Решение за $4^n$ запросов выглядит так: спросить про все листья, а потом посчитать на дереве динамику на выигрыш-проигрыш:

$$t_v = \begin{cases} 1 & \text{если } t_{v_l} = 0 \lor t_{v_r} = 0 \\ 0 & \text{иначе} \end{cases}$$

Это решение можно соптимизировать. Заметим, что если мы нашли переход из вершины в проигрышного сына, то второго сына можно не рассматривать, поэтому какие-то листья мы можем просто не посещать.

Введем $t_n$ — матожидание количества посещенных листьев для глубины $2n$.

Для начала рассмотрим $t_1$ и все шесть конфигураций:

| Листья | матожидание |
|---|---|
| $[0, 0, 0, 0]$ | 2 |
| $[0, 0, 0, 1]$ | 2.5 |
| $[0, 1, 0, 1]$ | 3 |
| $[0, 0, 1, 1]$ | 2.5 |
| $[0, 1, 1, 1]$ | 2.75 |
| $[1, 1, 1, 1]$ | 3 |

Заметим, что за каждый спуск на два уровня вниз, мы рассматриваем в среднем не более трех детей. Тогда $t_n \leq 3 \cdot t_{n-1} \leq 3^n$

**$\alpha\beta$-отсечение.** Проведем предыдущее рассуждение на минимаксной игре (это такая, где в листьях записаны числа, первый игрок хочет минимизировать итоговое число, а второй максимизировать). Введем параметры $\alpha$ и $\beta$ — гарантии игроков. $\beta$ — это минимальное число, которое первый игрок может себе гарантировать (то есть первый игрок знает, что не получит больше, чем $\beta$). Аналогично $\alpha$ — это максимальное число, которое гарантирует себе второй игрок). У нас должен соблюдаться инвариант $\alpha \leq \beta$, потому что остальные состояния неиграбельные. Как в них можно попасть по ходу перебора? Если в какой-то момент один из игроков сделает невыгодный для себя ход. Понятно, что при оптимальной игре обоих игроков они не будут делать невыгодные для себя ходы.

Как пересчитываются $\alpha$ и $\beta$? В листе $\alpha = \beta = get(v)$. Если второй игрок может пойти в состояние со стоимостью $x$, то он может сделать $\alpha = max(\alpha, x)$. Аналогично первый игрок будет уменьшать $\beta$. При этом эти гарантии будут переходить в сыновей вершины, но не в предков — в предки будут переходить только значение вершины (которое мы либо явно посчитали, либо вообще не сччитали, потому что состояние было неиграбельно).