Розбір задачі «1 раунд»

Достатньо знайти старший біт результату гри. Якщо старший біт числа має номер i, тоді число лежить в межах $[2^i, 2^{i+1}]$.

Переберемо, скільки старших бітів Арчі може зробити нулями.

Помітимо, що для перевірки того, що Арчі може занулити біти старші за i неважливі всі біти, які $\leqslant i$. Поділимо всі числа на 2^i.

Якщо на початку ходу Баріка xor масиву не рівен нулю, то він відразу може припинити гру. Отже, після кожного ходу Арчі xor масиву має бути рівен нулю. Арчі має тільки один варіант першого ходу, а далі повинен повторювати кроки Баріка. Звідси випливає, що або xor масиву з самого початку був рівен нулю і Арчі відразу припиняє гру, або кількість входжень рівно одного числа масиву (не враховуючи 0) має бути непарна.

Розбір задачі «2 раунд»

Розглянемо для початку наступну гру.

Поле, в клітинках якого стоять камінці. За один крок гравець вибирає квадрат, сторона якого не більше за к, забирає камінь з правої нижньої клітинки і ставить по камню у всі інші клітинки квадрата.

Якщо взяти конкретний стан гри, то можна представити його, як гру на багатьох полях в кожному з яких по одному камню. Отримати відповідь на таку задачу можна, використавши функцію Шпрага-Гранді. З цього випливає, що в будь-який момент нам неважлива кількість камінців в клітинці, а тільки її парність. Отже ці дві гри еквівалентні.

Згенерувавши відповіді для малих тестів можна з'ясувати, що функція Шпрага-Гранді для поля на якому зайнята тільки клітинка з координатами (i, j) рівна $min(lowestBit(i), lowestBit(j), greatestBit(k))$. Тут $lowestBit(i)$ - найбільша степінь двійки, що є дільником i, а $greatestBit(k)$ - максимальна степінь двійки, що не більша за k.

Залишилось тільки порахувати ксор цієї функції для всіх клітинок, що спочатку пофарбовані в білий. Порахуємо спочатку масив a, де a_i для $i \leqslant k$ - кількість білих клітинок обидві координати яких діляться без залишку на 2^i і 0 в іншому випадку. $a_i - a_{i-1}$ - кількість клітинок, для яких функція Шпрага-гранді рівна 2^i.

Розбір задачі «3 раунд»

Використаємо бінарний пошук.

Щоб перевірити чи відповідь більша за a числа на листках, що більші за a замінимо на 1, а всі інші на 0. Для розв'язання задачі необхідно сказати, чи Арчі зможе отримати 1 в кінці гри.

Назвемо піддерево парним, якщо в ньому можна зробити парну кількість кроків.

Якщо обидва піддерева синів кореня парні, то Арчі достатньо виграти хоча б в одному, адже він буде робити останній крок і може вибрати потрібну вершину. Помітимо піддерево в якому виграє Арчі як 1, а інше як 2. Перший крок Арчі робить в піддереві 1, далі якщо Барік робить крок в піддереві 2, то Арчі робить випадковий крок в тому ж піддереві, а якщо Барік робить крок в піддереві 1, то Арчі робить оптимальний крок в цьому ж піддереві(за припущенням воно виграшне).

Якщо одне піддерево парне, а інше непарне, то Арчі повинен вигравати в обох. Складним є випадок коли розміри обох піддерев непарні. В такому випадку Арчі не завжди може зробити випадковий крок в піддереві 2. Через це в Баріка з'являється можливість 1 раз пропустити крок. Напишемо динаміку на дереві станами якої буде вершина, гравець і чи є можливість пропустити крок. Цієї інформації цілком достатньо, щоб її перерахувати.

Випадок, коли два піддерева парні і не можна пропустити хід. Гравець виграє тільки, якщо він виграє хоча б в одному піддереві без можливості пропуску ходу.

Випадок, коли два піддерева непарні і не можна пропустити хід. Гравець виграє, якщо він виграє хоча б в одному піддереві з можливістю пропустити хід.

Випадок, коли піддерева мають різну парність і не можна пропустити хід. Якщо ми виграєм в непарному дереві і інший гравець програє в парному без можливості пропуску, то ми ходимо в непарне дерево і виграємо гру. Якщо ми виграєм в парному з можливістю пропустити хід і інший гравець програє в непарному з можливістю пропустити хід, то ми робимо хід в парному і інший гравець попадає в попередній випадок, в якому він програє.

Випадок, коли піддерева парні і можна пропустити хід. Ми можемо пропустити хід, або піти грати в одне піддерево з можливістю пропуску ходу. В другому випадку ми виграєм, якщо можем виграти в одному піддереві з пропуском ходу і наш суперник не може виграти в іншому без пропуску.

Випадок, коли піддерева непарні і можна пропустити хід. Аналогічно до минулого випадку.

Випадок, коли піддерева різної парності і можна пропустити крок.Нам потрібно вигравати або парне дерево без пропуску, або непарне з пропуском.

Ідея задачі взята звідси: https://codeforces.com/problemset/problem/457/F

Розбір задачі «4 раунд»

Рекурсивно порахувати функцію Шпрага-Гранді для станів, що характеризуються розмірами куба. Якщо в нас є куб розмірами $a_1, a_2, ..., a_n$ то відповідь для нього це mex по всім коректним способам вибрати клітинку розбиття xor відповідей всіх кубів, на який розпадається наш куб при цьому розбитті.

Розбір задачі «5 раунд»

Якщо походити крайньою фігурою, то гра перейде до аналогічною для дошки розміру $n - 2$.

Якщо походити середньою фігурою з номером i, то гра розіб'ється на дві гри для дошок розміром $n - i - 1$ і $i - 2$.

Відповідь легко порахувати, як функцію Шпрага-Гранді для малих значень. Далі можна помітити, що послідовність відповідей з певного моменту стає періодичною з періодом 34.

Розбір задачі «6 раунд»

Спочатку рухаєм перший справа білий камінь вперед. Потім 2 лівих чорних каменя. Далі 3 правих чорних каменя і т. д. Коли найправіший білий камінь впреться в кінець про нього можна забути і рухати всі інші камені за ним по черзі. Аналогічно з чорними.

Розбір задачі «7 раунд»

Розглянемо 2 параметри - парність кількості рядків, які можна утворити перемішуванням заданого і які не зустрічались раніше(позначимо це число, як) та парність n. Гра завжди закінчується у випадку, коли обидва з них парні (пустий рядок). З ситуації коли n непарне завжди можна перейти у ситуацію коли парне і a і n. Якщо n парне, а a - непарне, то ми перемішуванням переходимо у програшну ситуацію.

Звідси виходить, що Арчі програє тільки тоді, коли n - парне, і стрічка має непарну кількість різних перестановок. Це число можна порахувати за допомогою динаміки.

Потрібно знайти кількість способів розбити число n на доданки $a_1, a_2, ..., a_m$ таким чином, щоб найбільша степінь двійки на яку ділиться $n!$ була рівна найбільшій степені двійки, на яку ділиться $a_1! \cdot a_2! \cdot ... \cdot a_m!$. Найбільша степінь двійки, на яку ділиться $x!$ рівна $x - bits(x)$, де $bits(x)$ - кількість одиничних бітів в числі x. розписавши формули можна отримати, що $a_1 + a_2 + ... + a_m = n$ і $bits(a_1) + bits(a_2) + ... + bits(a_m) = bits(n)$. звідси випливає, що $a_1|a_2|...|a_m = n$. Напишемо динаміку, першим параметром якої буде маска n, яку ми зібрали, а другим - кількість чисел які використали. Складність $O(k * 3^{log(n)})$.

Розбір задачі «8 раунд»

Щоразу одна сторона стрічки буде знищуватися, а інша ділитись пополам. Можна побудувати дерево вершинами якого будуть стани гри і, написавши на ньому тривіальну динаміку, знайти відповідь.

Розбір задачі «9 раунд»

Нехай $n \leqslant m$. Якщо $n = 1$, то відповідь $(m + 1)/2$.

Інакше відповідь - або 1 або 2 (за допомогою одного каменя можна видалити три, що стоять над ним).