Теория игр: числа Гранди

Борис Золотов (Б06)
Иван Казменко (Б01, Б05)
Владислав Макаров (Б02, Б03)
Вячеслав Соколов (Б09)
Арина Филимонова (Б10)

Санкт-Петербургский Государственный Университет

Вторник, 26 сентября 2023 года

Содержание

Правила

Общие правила всех рассматриваемых игр:

- два игрока

Правила

Общие правила всех рассматриваемых игр:

- два игрока
- ходят по очереди

Правила

Общие правила всех рассматриваемых игр:

- два игрока
- ходят по очереди
- проигрывает тот, кто не может ходить

Правила

Общие правила всех рассматриваемых игр:

- два игрока
- ходят по очереди
- проигрывает тот, кто не может ходить
- правила хода одинаковые для обоих (беспристрастная или нейтральная игра)

Правила

Общие правила всех рассматриваемых игр:

- два игрока
- ходят по очереди
- проигрывает тот, кто не может ходить
- правила хода одинаковые для обоих (беспристрастная или нейтральная игра)
- все всё знают (игра с полной информацией)

Правила

Общие правила всех рассматриваемых игр:

- два игрока
- ходят по очереди
- проигрывает тот, кто не может ходить
- правила хода одинаковые для обоих (беспристрастная или нейтральная игра)
- все всё знают (игра с полной информацией)
- игра конечная (как бы ни ходили игроки, за конечное число ходов кто-то обязательно проиграет)

Правила

Общие правила всех рассматриваемых игр:

- два игрока
- ходят по очереди
- проигрывает тот, кто не может ходить
- правила хода одинаковые для обоих (беспристрастная или нейтральная игра)
- все всё знают (игра с полной информацией)
- игра конечная (как бы ни ходили игроки, за конечное число ходов кто-то обязательно проиграет)
- вопрос: кто выиграет *при правильной игре*?

Пример 1

Пример 1:

- кучка из n камней

Пример 1

Пример 1:

- кучка из n камней
- за ход можно взять 1, 2 или 3 камня

Пример 1

Пример 1:

- кучка из n камней
- за ход можно взять 1, 2 или 3 камня
- кто победит?

Пример 1

Пример 1:

- кучка из n камней
- за ход можно взять 1, 2 или 3 камня
- кто победит?
- решение: рассмотрим $n \bmod 4$

Пример 1

Пример 1:

- кучка из n камней
- за ход можно взять 1, 2 или 3 камня
- кто победит?
- решение: рассмотрим $n \bmod 4$
- $n \bmod 4 \neq 0$: сделаем $= 0$

Пример 1

Пример 1:

- кучка из n камней
- за ход можно взять 1, 2 или 3 камня
- кто победит?
- решение: рассмотрим $n \bmod 4$
- $n \bmod 4 \neq 0$: сделаем $= 0$
- $n \bmod 4 = 0$: придётся сделать $\neq 0$

Общие идеи

Общие идеи при решении:

- каждая позиция либо выигрышная, либо проигрышная

Общие идеи

Общие идеи при решении:

- каждая позиция либо выигрышная, либо проигрышная
- неважно, чей ход

Общие идеи

Общие идеи при решении:

- каждая позиция либо выигрышная, либо проигрышная
- неважно, чей ход
- тот, кто ходит, либо проигрывает, либо выигрывает

Пример 2

Пример 2:

- кучка из n камней

Пример 2

Пример 2:

- кучка из n камней
- за ход можно взять 1, 3 или 5 камней

Пример 2

Пример 2:

- кучка из n камней
- за ход можно взять 1, 3 или 5 камней
- кто победит?

Пример 2

Пример 2:

- кучка из n камней
- за ход можно взять 1, 3 или 5 камней
- кто победит?
- решение: чётность

Пример 2

Пример 2:

- кучка из n камней
- за ход можно взять 1, 3 или 5 камней
- кто победит?
- решение: чётность
- от ходов не зависит исход

Пример 3

Пример 3:

- кучка из n камней

Пример 3

Пример 3:

- кучка из n камней
- за ход можно взять 2, 3 или 5 камней

Пример 3

Пример 3:

- кучка из n камней
- за ход можно взять 2, 3 или 5 камней
- кто победит?

Пример 3

Пример 3:

- кучка из n камней
- за ход можно взять 2, 3 или 5 камней
- кто победит?
- общий случай

Пример 3

Пример 3:

- кучка из n камней
- за ход можно взять 2, 3 или 5 камней
- кто победит?
- общий случай
- позиции: 0, 1, ...

Пример 3

Пример 3:

- кучка из n камней
- за ход можно взять 2, 3 или 5 камней
- кто победит?
- общий случай
- позиции: 0, 1, ...
- каждая позиция — либо выигрышная, либо проигрышная

Пример 3

Пример 3:

- кучка из n камней
- за ход можно взять 2, 3 или 5 камней
- кто победит?
- общий случай
- позиции: 0, 1, ...
- каждая позиция — либо выигрышная, либо проигрышная
- В — выигрыш, П — проигрыш

Пример 3

Пример 3:

- кучка из n камней
- за ход можно взять 2, 3 или 5 камней
- кто победит?
- общий случай
- позиции: 0, 1, ...
- каждая позиция — либо выигрышная, либо проигрышная
- В — выигрыш, П — проигрыш
- П $\Leftrightarrow$ нет ходов в П

Пример 3

Пример 3:

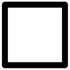
- кучка из n камней
- за ход можно взять 2, 3 или 5 камней
- кто победит?
- общий случай
- позиции: 0, 1, ...
- каждая позиция — либо выигрышная, либо проигрышная
- В — выигрыш, П — проигрыш
- П $\Leftrightarrow$ нет ходов в П
- В $\Leftrightarrow$ есть ход в П

Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней

Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



0

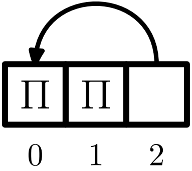
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней

П	
0	1

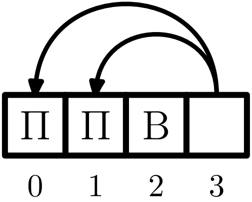
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



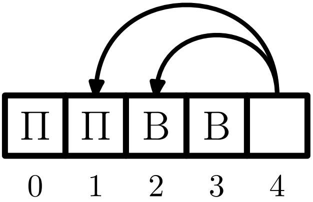
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



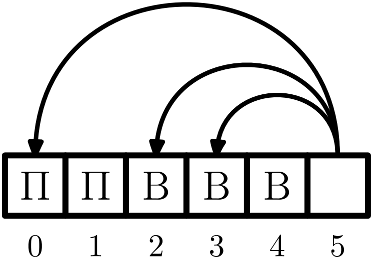
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



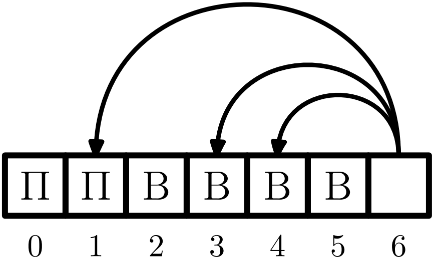
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



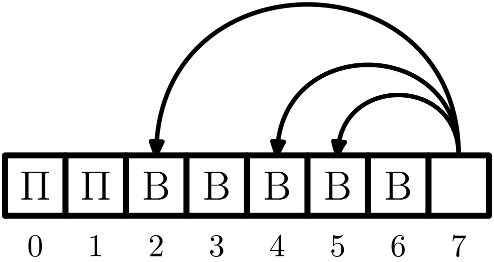
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



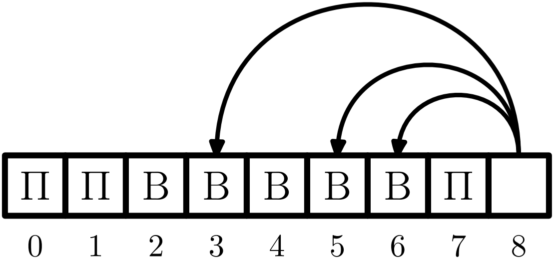
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



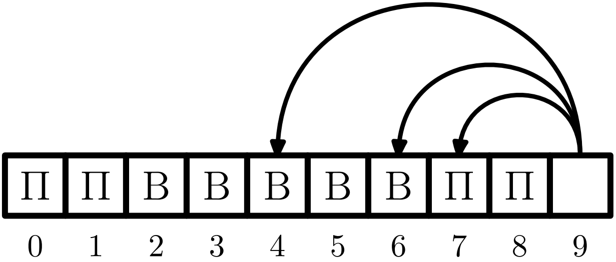
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



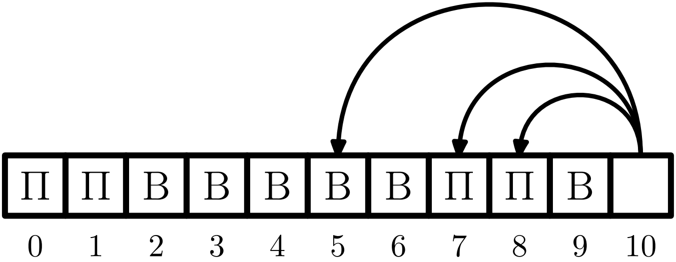
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



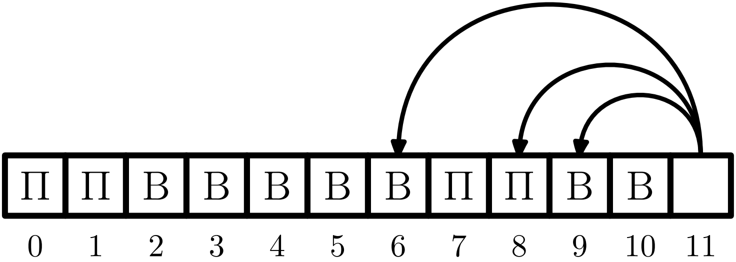
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



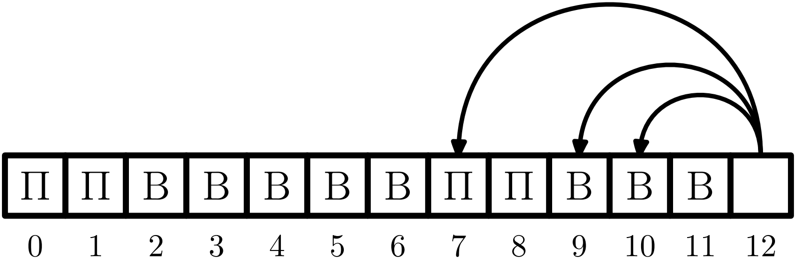
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



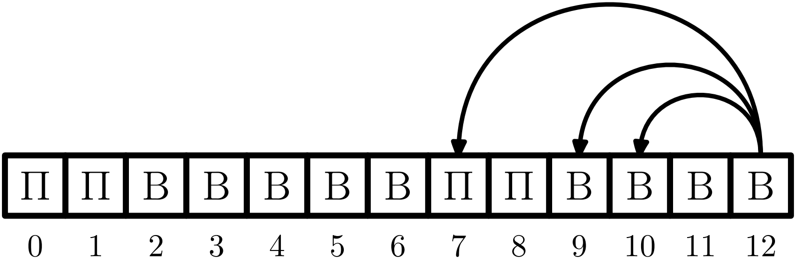
Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней



Пример 3

Пример 3: за ход можно взять 2, 3 или 5 камней

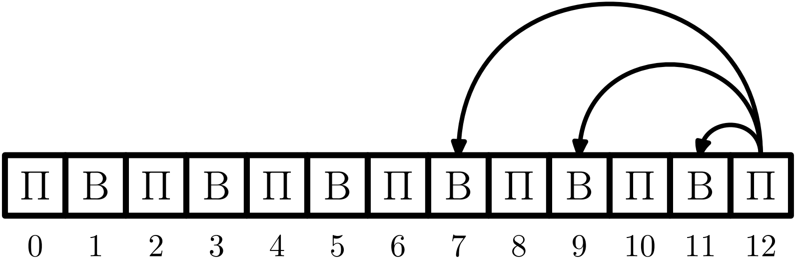


Пример 2

Пример 2: за ход можно взять 1, 3 или 5 камней

Пример 2

Пример 2: за ход можно взять 1, 3 или 5 камней

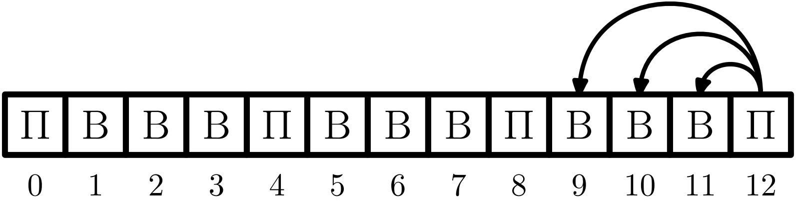


Пример 1

Пример 1: за ход можно взять 1, 2 или 3 камня

Пример 1

Пример 1: за ход можно взять 1, 2 или 3 камня



Код

Пример 1: за ход можно взять 1, 2 или 3 камня

```cpp
#include <iostream>
using namespace std;
int main () {
    int n;
    cin >> n;
    cout << ((n % 4 == 0) ? "Second" : "First") << endl;
    return 0;
}
```

Код

Пример 1: за ход можно взять 1, 2 или 3 камня

```cpp
#include <iostream>
using namespace std;
int main () {
    int n;
    cin >> n;
    if (n % 4 == 0)
        cout << -1 << endl;
    else
        while (true) {
            cout << n % 4 << endl;

            n -= n % 4;
            int v;
            cin >> v;
            if (v == -1)
                break;
            n -= v;
        }
    return 0;
}
```

Код

Пример 1: за ход можно взять 1, 2 или 3 камня

```
#include <cstdio>

int main () {
    int n;
    scanf ("%d", &n);
    if (n % 4 == 0)
        printf ("-1\n");
    else
        while (true) {
            printf ("%d\n", n % 4);
            fflush (stdout);
            n -= n % 4;
            int v;
            scanf ("%d", &v);
            if (v == -1)
                break;
            n -= v;
        }
    return 0;
}
```

Код

Пример 3: за ход можно взять 2, 3 или 5 камней

```
#include <iostream>
#include <vector>
using namespace std;
vector <int> moves = {2, 3, 5};
int main () {
    int n;
    cin >> n;
    vector <bool> win (n + 1);
    for (int i = 0; i <= n; i++)
        for (auto k : moves)
            if (i >= k && !win[i - k])
                win[i] = true;
    cout << (win[n] ? "First" : "Second") << endl;
    return 0;
}
```

Игра «Ним»

Игра «Ним»:

- кучка из n камней

Игра «Ним»

Игра «Ним»:

- кучка из n камней
- за ход можно взять любое положительное число камней

Игра «Ним»

Игра «Ним»:

- кучка из n камней
- за ход можно взять любое положительное число камней
- кто победит?

Игра «Ним»

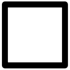
Игра «Ним»:

- кучка из n камней
- за ход можно взять любое положительное число камней
- кто победит?
- первый (если кучка не пуста)

Игра «Ним»

Игра «Ним»:

- кучка из n камней
- за ход можно взять любое положительное число камней
- кто победит?
- первый (если кучка не пуста)

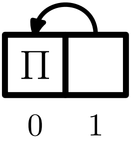


0

Игра «Ним»

Игра «Ним»:

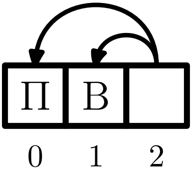
- кучка из n камней
- за ход можно взять любое положительное число камней
- кто победит?
- первый (если кучка не пуста)



Игра «Ним»

Игра «Ним»:

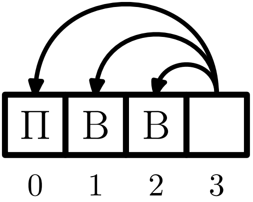
- кучка из n камней
- за ход можно взять любое положительное число камней
- кто победит?
- первый (если кучка не пуста)



Игра «Ним»

Игра «Ним»:

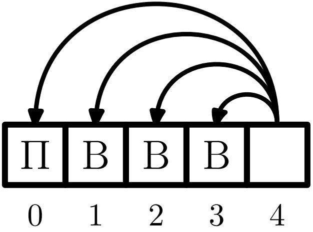
- кучка из n камней
- за ход можно взять любое положительное число камней
- кто победит?
- первый (если кучка не пуста)



Игра «Ним»

Игра «Ним»:

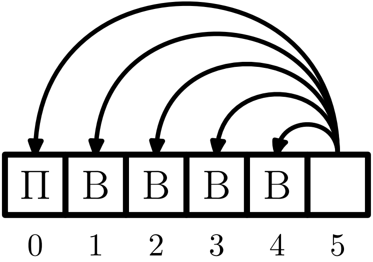
- кучка из n камней
- за ход можно взять любое положительное число камней
- кто победит?
- первый (если кучка не пуста)



Игра «Ним»

Игра «Ним»:

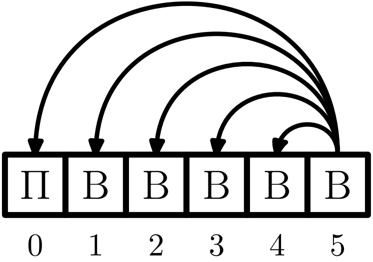
- кучка из n камней
- за ход можно взять любое положительное число камней
- кто победит?
- первый (если кучка не пуста)



Игра «Ним»

Игра «Ним»:

- кучка из n камней
- за ход можно взять любое положительное число камней
- кто победит?
- первый (если кучка не пуста)



Комбинация игр

Скомбинируем две такие игры:

- две кучки: a и b камней

Комбинация игр

Скомбинируем две такие игры:

- две кучки: a и b камней
- за ход можно взять любое положительное число камней, но только из одной кучки

Комбинация игр

Скомбинируем две такие игры:

- две кучки: a и b камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?

Комбинация игр

Скомбинируем две такие игры:

- две кучки: a и b камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?
- симметричная стратегия

Комбинация игр

Скомбинируем две такие игры:

- две кучки: a и b камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?
- симметричная стратегия
- если $a \neq b$, можно сделать их равными

Комбинация игр

Скомбинируем две такие игры:

- две кучки: a и b камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?
- симметричная стратегия
- если $a \neq b$, можно сделать их равными
- если $a = b$, придётся сделать их различными

Комбинация игр

Скомбинируем две такие игры:

- две кучки: a и b камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?
- симметричная стратегия
- если $a \neq b$, можно сделать их равными
- если $a = b$, придётся сделать их различными
- если $a = b = 0$, и этого сделать не получится

Двумерная таблица

То же самое можно посчитать, как в общем случае:

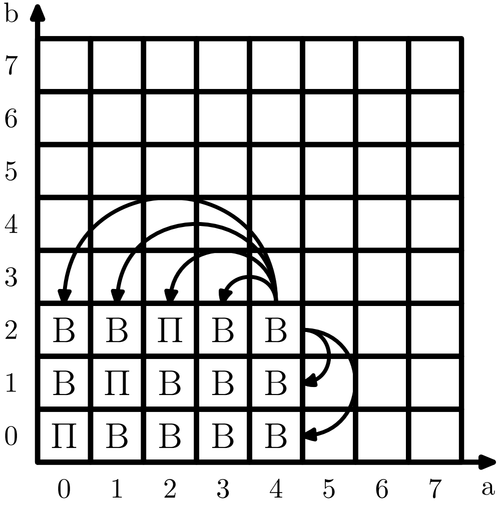
- каждой паре (a, b) соответствует клетка в таблице

Двумерная таблица

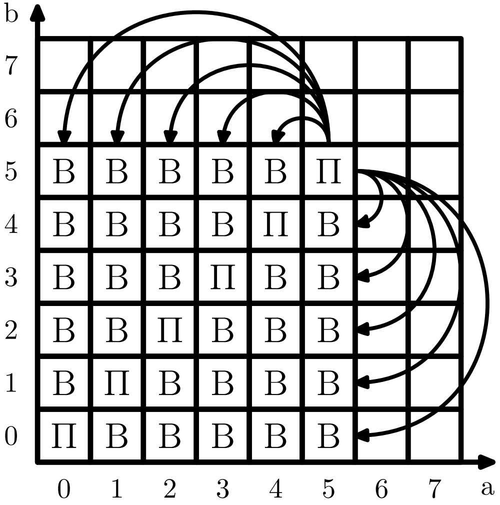
То же самое можно посчитать, как в общем случае:

- каждой паре (a, b) соответствует клетка в таблице
- из клетки (a, b) можно сделать ход в $(a-1, b)$, $(a-2, b)$, ... и в $(a, b-1)$, $(a, b-2)$, ...

Двумерная таблица



Двумерная таблица



Двумерная таблица

b \ a	0	1	2	3	4	5	6	7
7	В	В	В	В	В	В	В	П
6	В	В	В	В	В	В	П	В
5	В	В	В	В	В	П	В	В
4	В	В	В	В	П	В	В	В
3	В	В	В	П	В	В	В	В
2	В	В	П	В	В	В	В	В
1	В	П	В	В	В	В	В	В
0	П	В	В	В	В	В	В	В

Три кучки

Что, если скомбинировать три игры?

- три кучки: a, b и c камней

Три кучки

Что, если скомбинировать три игры?

- три кучки: a, b и c камней
- за ход можно взять любое положительное число камней, но только из одной кучки

Три кучки

Что, если скомбинировать три игры?

- три кучки: a, b и c камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?

Три кучки

Что, если скомбинировать три игры?

- три кучки: a, b и c камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?
- можно построить трёхмерную таблицу

Три кучки

Что, если скомбинировать три игры?

- три кучки: a, b и c камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?
- можно построить трёхмерную таблицу
- хочется решить более эффективно

Пример

Пример: $a = 1$, $b = 2$, $c = 3$.

Пример

Пример: $a = 1$, $b = 2$, $c = 3$.

- $(1, 2, 3) \to (0, 2, 3) \to (0, 2, 2)$

Пример

Пример: $a = 1$, $b = 2$, $c = 3$.

- $(1, 2, 3) \to (0, 2, 3) \to (0, 2, 2)$
- $(1, 2, 3) \to (1, 1, 3) \to (1, 1, 0)$

Пример

Пример: $a = 1$, $b = 2$, $c = 3$.

- $(1, 2, 3) \to (0, 2, 3) \to (0, 2, 2)$
- $(1, 2, 3) \to (1, 1, 3) \to (1, 1, 0)$
- $(1, 2, 3) \to (1, 0, 3) \to (1, 0, 1)$

Пример

Пример: $a = 1$, $b = 2$, $c = 3$.

- $(1, 2, 3) \to (0, 2, 3) \to (0, 2, 2)$
- $(1, 2, 3) \to (1, 1, 3) \to (1, 1, 0)$
- $(1, 2, 3) \to (1, 0, 3) \to (1, 0, 1)$
- $(1, 2, 3) \to (1, 2, 2) \to (0, 2, 2)$

Пример

Пример: $a = 1$, $b = 2$, $c = 3$.

- $(1, 2, 3) \to (0, 2, 3) \to (0, 2, 2)$
- $(1, 2, 3) \to (1, 1, 3) \to (1, 1, 0)$
- $(1, 2, 3) \to (1, 0, 3) \to (1, 0, 1)$
- $(1, 2, 3) \to (1, 2, 2) \to (0, 2, 2)$
- $(1, 2, 3) \to (1, 2, 1) \to (1, 0, 1)$

Пример

Пример: $a = 1$, $b = 2$, $c = 3$.

- $(1, 2, 3) \to (0, 2, 3) \to (0, 2, 2)$
- $(1, 2, 3) \to (1, 1, 3) \to (1, 1, 0)$
- $(1, 2, 3) \to (1, 0, 3) \to (1, 0, 1)$
- $(1, 2, 3) \to (1, 2, 2) \to (0, 2, 2)$
- $(1, 2, 3) \to (1, 2, 1) \to (1, 0, 1)$
- $(1, 2, 3) \to (1, 2, 0) \to (1, 1, 0)$

Пример

Пример: $a = 1$, $b = 2$, $c = 3$.

- $(1, 2, 3) \to (0, 2, 3) \to (0, 2, 2)$
- $(1, 2, 3) \to (1, 1, 3) \to (1, 1, 0)$
- $(1, 2, 3) \to (1, 0, 3) \to (1, 0, 1)$
- $(1, 2, 3) \to (1, 2, 2) \to (0, 2, 2)$
- $(1, 2, 3) \to (1, 2, 1) \to (1, 0, 1)$
- $(1, 2, 3) \to (1, 2, 0) \to (1, 1, 0)$
- получается, что любой ход ведёт к проигрышу

Другие примеры

Другие примеры:

- $(24, 47, 55)$ — проигрышный

Другие примеры

Другие примеры:

- $(24, 47, 55)$ — проигрышный
- $(23, 48, 55)$ — выигрышный

Другие примеры

Другие примеры:

- $(24, 47, 55)$ — проигрышный
- $(23, 48, 55)$ — выигрышный
- чтобы научиться быстро решать, изучим теорию

Игра на графе

Обобщение — игра на графе:

- есть ориентированный ациклический конечный граф

Игра на графе

Обобщение — игра на графе:

- есть ориентированный ациклический конечный граф
- в одной из вершин стоит фишка

Игра на графе

Обобщение — игра на графе:

- есть ориентированный ациклический конечный граф
- в одной из вершин стоит фишка
- двое ходят по очереди

Игра на графе

Обобщение — игра на графе:

- есть ориентированный ациклический конечный граф
- в одной из вершин стоит фишка
- двое ходят по очереди
- за ход можно подвинуть фишку по дуге

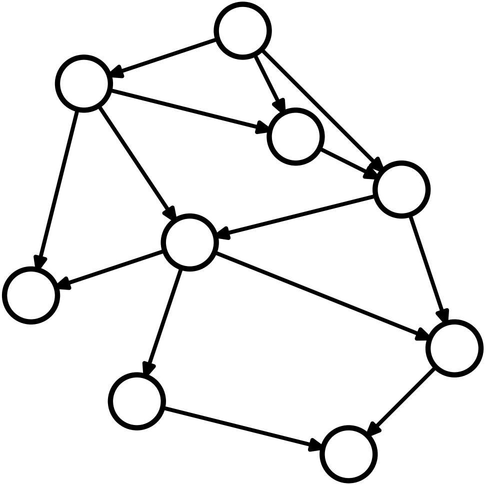
Игра на графе

Обобщение — игра на графе:

- есть ориентированный ациклический конечный граф
- в одной из вершин стоит фишка
- двое ходят по очереди
- за ход можно подвинуть фишку по дуге
- проигрывает тот, кто не может сделать ход

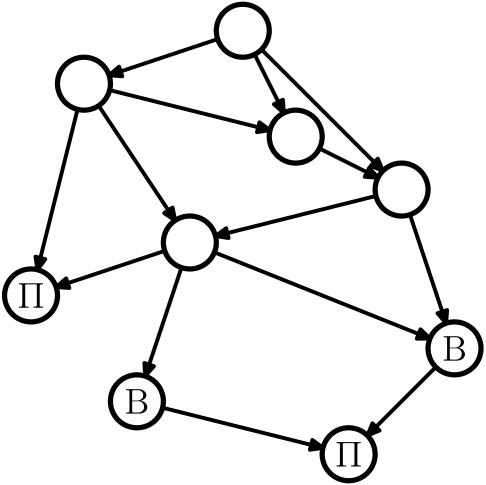
Пример

Пример:



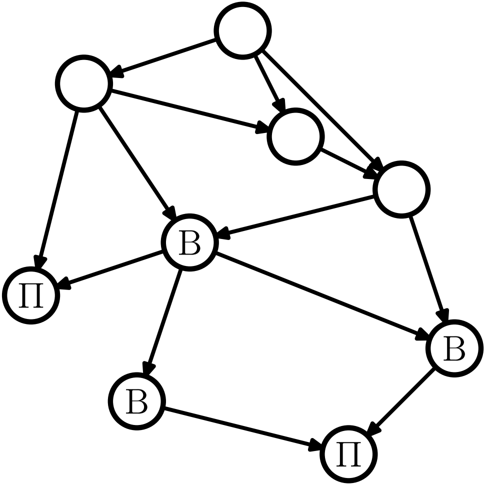
Пример

Пример:



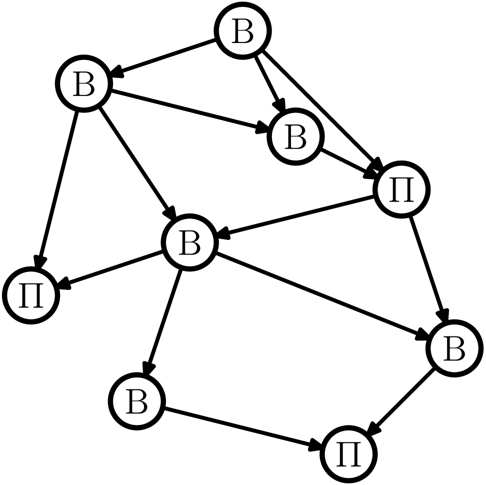
Пример

Пример:



Пример

Пример:



Решение

Решение — заполняем вершины по индукции:

- если нет ходов, то П

Решение

Решение — заполняем вершины по индукции:

- если нет ходов, то П
- если есть ход в П, то В

Решение

Решение — заполняем вершины по индукции:

- если нет ходов, то П
- если есть ход в П, то В
- если все ходы в В, то П

Две фишки

Определение 1: игра — это граф и фишка на нём.

Две фишки

Определение 1: игра — это граф и фишка на нём.

- что, если поставить две фишки?

Две фишки

Определение 1: игра — это граф и фишка на нём.

- что, если поставить две фишки?
- за ход можно подвинуть одну фишку по дуге

Две фишки

Определение 1: игра — это граф и фишка на нём.

- что, если поставить две фишки?
- за ход можно подвинуть одну фишку по дуге
- можно два разных графа с фишками нарисовать рядом, получится один граф с двумя фишками

Две фишки

Определение 1: игра — это граф и фишка на нём.

- что, если поставить две фишки?
- за ход можно подвинуть одну фишку по дуге
- можно два разных графа с фишками нарисовать рядом, получится один граф с двумя фишками

Определение 1′: игра — это граф и набор фишек на нём; ход в этом объединении — подвинуть какую-то одну фишку по ребру.

Определение 2: прямая сумма игр — это объединение графов и наборов фишек на них.

Две фишки

Определение 1: игра — это граф и фишка на нём.

- что, если поставить две фишки?
- за ход можно подвинуть одну фишку по дуге
- можно два разных графа с фишками нарисовать рядом, получится один граф с двумя фишками

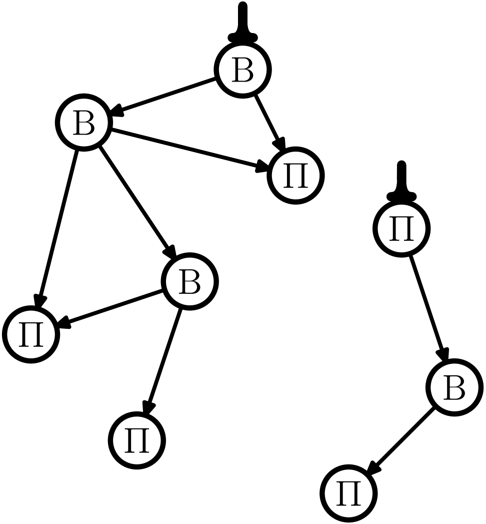
Определение 1′: игра — это граф и набор фишек на нём; ход в этом объединении — подвинуть какую-то одну фишку по ребру.

Определение 2: прямая сумма игр — это объединение графов и наборов фишек на них.

Обозначение: $A \oplus B$ — это прямая сумма игр A и B.

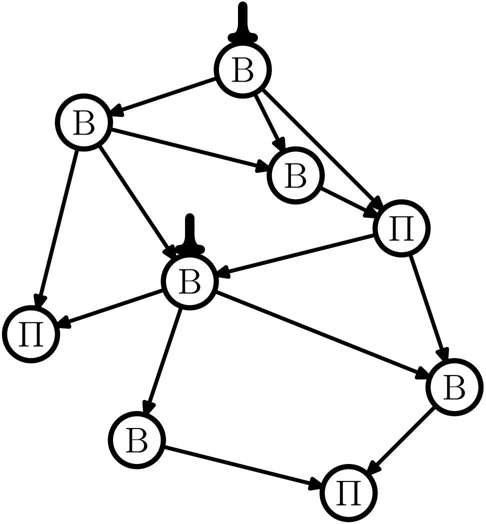
Пример

Пример:



Пример

Кто победит?



Минимальное исключённое

Видно, что информации «В или П» недостаточно.

Минимальное исключённое

Видно, что информации «В или П» недостаточно.
Напишем в вершинах целые неотрицательные числа.

Минимальное исключённое

Видно, что информации «В или П» недостаточно.
Напишем в вершинах целые неотрицательные числа.

Определение 3: Минимальное исключённое (minimal excludant) — функция от множества чисел: минимальное целое неотрицательное число, которое не встречается в этом множестве.

Минимальное исключённое

Видно, что информации «В или П» недостаточно.
Напишем в вершинах целые неотрицательные числа.

Определение 3: Минимальное исключённое (minimal excludant) — функция от множества чисел: минимальное целое неотрицательное число, которое не встречается в этом множестве.

$\mathrm{mex}(S) = \mathsf{min}\{x \in \mathbb{N}_0 \mid x \notin S\}$

Минимальное исключённое

Видно, что информации «В или П» недостаточно.
Напишем в вершинах целые неотрицательные числа.

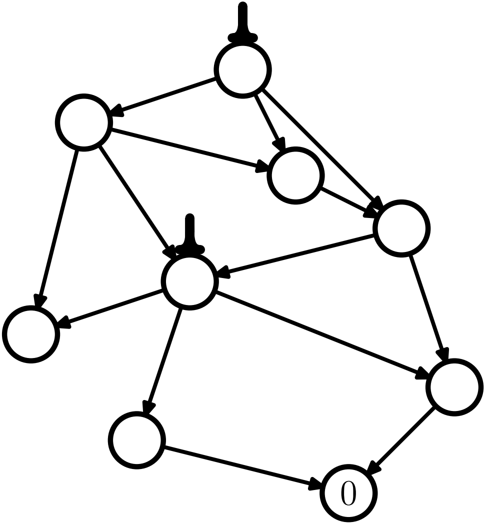
Определение 3: Минимальное исключённое (minimal excludant) — функция от множества чисел: минимальное целое неотрицательное число, которое не встречается в этом множестве.

$\text{mex}(S) = \mathsf{min}\{x \in \mathbb{N}_0 \mid x \notin S\}$

Определение 4: Число Гранди (Grundy number) в вершине — это mex от множества чисел Гранди во всех вершинах, куда из неё можно сделать ход.

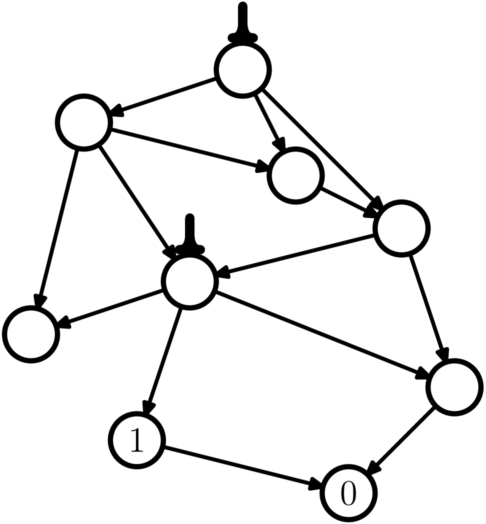
Пример вычисления

Вычисление mex:



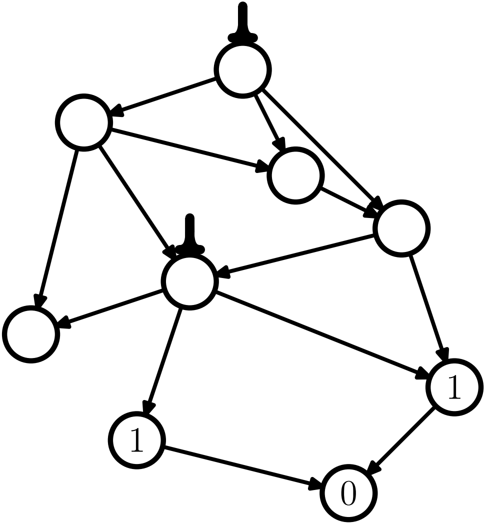
Пример вычисления

Вычисление mex:



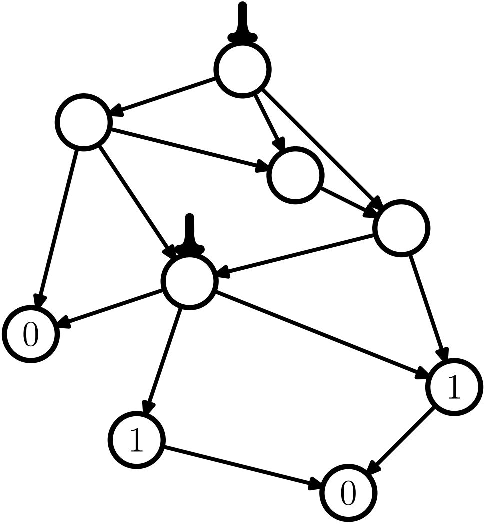
Пример вычисления

Вычисление mex:



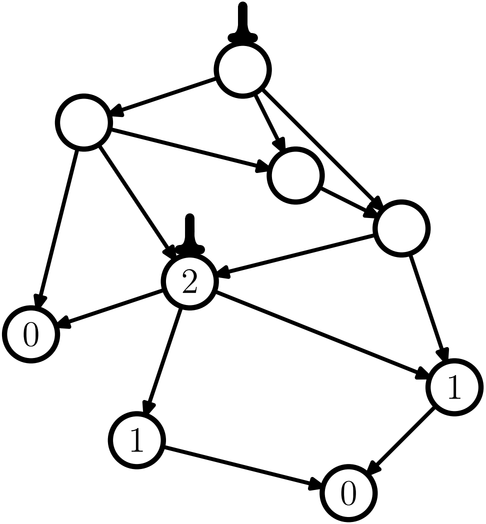
Пример вычисления

Вычисление mex:



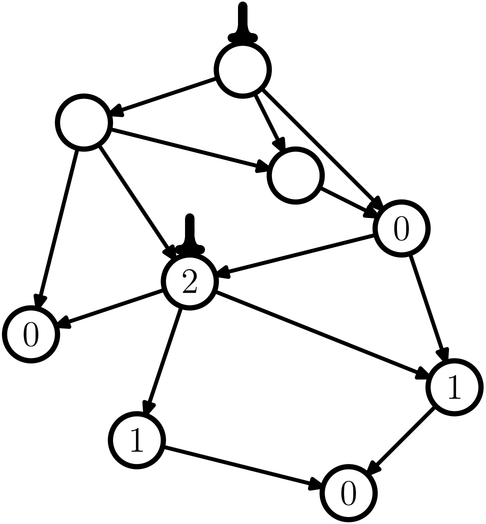
Пример вычисления

Вычисление mex:



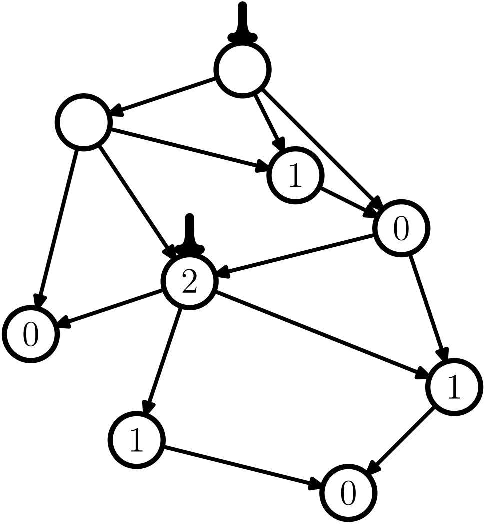
Пример вычисления

Вычисление mex:



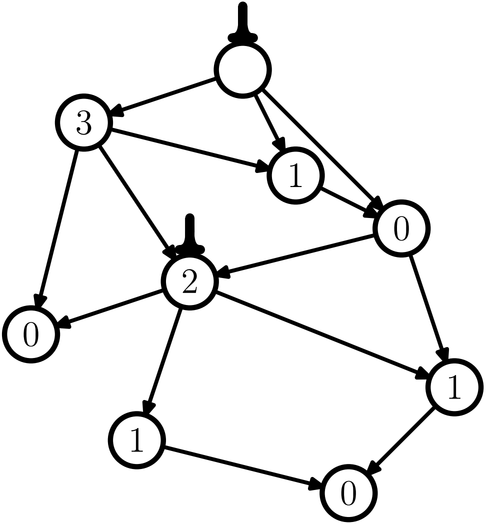
Пример вычисления

Вычисление mex:



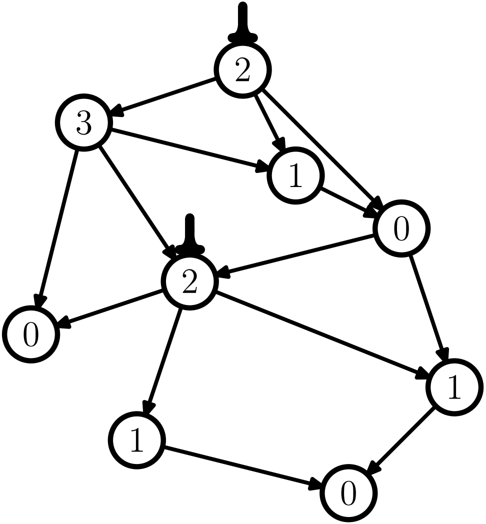
Пример вычисления

Вычисление mex:



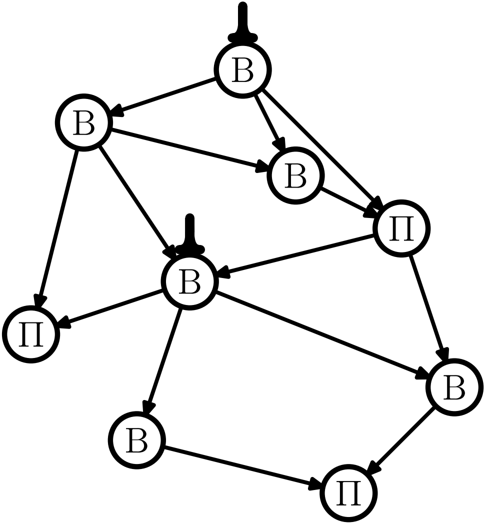
Пример вычисления

Вычисление mex:



Пример вычисления

Вычисление mex:



Числа Гранди и выигрыш

Утверждение 1: Число Гранди в вершине равно нулю $\Leftrightarrow$ эта вершина проигрышная.

Числа Гранди и выигрыш

Утверждение 1: Число Гранди в вершине равно нулю $\Leftrightarrow$ эта вершина проигрышная.

- можно доказать индукцией, пройдя по вершинам графа в порядке вычисления

Числа Гранди и выигрыш

Утверждение 1: Число Гранди в вершине равно нулю $\Leftrightarrow$ эта вершина проигрышная.

- можно доказать индукцией, пройдя по вершинам графа в порядке вычисления
- если нет ходов, то $0 \leftrightarrow \Pi$

Числа Гранди и выигрыш

Утверждение 1: Число Гранди в вершине равно нулю $\Leftrightarrow$ эта вершина проигрышная.

- можно доказать индукцией, пройдя по вершинам графа в порядке вычисления
- если нет ходов, то $0 \leftrightarrow \Pi$
- если есть ход в $0 \leftrightarrow \Pi$, то не-$0 \leftrightarrow$ В

Числа Гранди и выигрыш

Утверждение 1: Число Гранди в вершине равно нулю $\Leftrightarrow$ эта вершина проигрышная.

- можно доказать индукцией, пройдя по вершинам графа в порядке вычисления
- если нет ходов, то $0 \leftrightarrow \Pi$
- если есть ход в $0 \leftrightarrow \Pi$, то не-$0 \leftrightarrow$ В
- если все ходы в не-$0 \leftrightarrow$ В, то $0 \leftrightarrow \Pi$

Числа Гранди и выигрыш

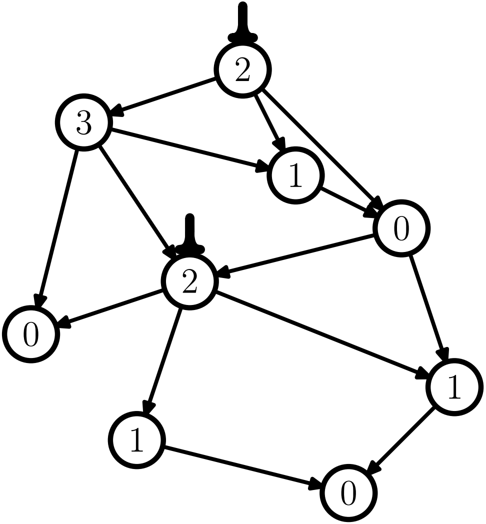
Утверждение 1: Число Гранди в вершине равно нулю $\Leftrightarrow$ эта вершина проигрышная.

- можно доказать индукцией, пройдя по вершинам графа в порядке вычисления
- если нет ходов, то $0 \leftrightarrow$ П
- если есть ход в $0 \leftrightarrow$ П, то не-$0 \leftrightarrow$ В
- если все ходы в не-$0 \leftrightarrow$ В, то $0 \leftrightarrow$ П

Итак, число Гранди в вершине содержит, в частности, информацию про выигрышность этой вершины.

Пример

Вернёмся к примеру:



Числа Гранди и две фишки

Утверждение 2: Игра с двумя фишками проигрышная $\Leftrightarrow$ числа Гранди под фишками равны.

Числа Гранди и две фишки

Утверждение 2: Игра с двумя фишками проигрышная $\Leftrightarrow$ числа Гранди под фишками равны.

- симметричная стратегия

Числа Гранди и две фишки

Утверждение 2: Игра с двумя фишками проигрышная $\Leftrightarrow$ числа Гранди под фишками равны.

- симметричная стратегия
- если числа под фишками не равны, можно сделать их равными, подвинув фишку с большего в число, равное меньшему — по определению mex такой ход существует

Числа Гранди и две фишки

Утверждение 2: Игра с двумя фишками проигрышная $\Leftrightarrow$ числа Гранди под фишками равны.

- симметричная стратегия
- если числа под фишками не равны, можно сделать их равными, подвинув фишку с большего в число, равное меньшему — по определению mex такой ход существует
- если числа под фишками равны, придётся сделать их различными — по определению mex нельзя сделать ход в такое же число

Числа Гранди и две фишки

Утверждение 2: Игра с двумя фишками проигрышная ⇔ числа Гранди под фишками равны.

- симметричная стратегия
- если числа под фишками не равны, можно сделать их равными, подвинув фишку с большего в число, равное меньшему — по определению mex такой ход существует
- если числа под фишками равны, придётся сделать их различными — по определению mex нельзя сделать ход в такое же число
- рано или поздно равные числа окажутся нулями, из которых ещё и нет ходов

Определение эквивалентности

Пусть A и B — две игры, каждая с одной фишкой.

Определение эквивалентности

Пусть A и B — две игры, каждая с одной фишкой.

Пусть $g(E)$ — число Гранди для игры E с одной фишкой: то число, которое стоит под фишкой.

Определение эквивалентности

Пусть A и B — две игры, каждая с одной фишкой.

Пусть $g(E)$ — число Гранди для игры E с одной фишкой: то число, которое стоит под фишкой.

Определение 5: Игры A и B эквивалентны, если для любой игры C результат прямой суммы $A \oplus C$ (выигрыш или проигрыш) такой же, как результат $B \oplus C$.

Определение эквивалентности

Пусть A и B — две игры, каждая с одной фишкой.

Пусть $g(E)$ — число Гранди для игры E с одной фишкой: то число, которое стоит под фишкой.

Определение 5: Игры A и B эквивалентны, если для любой игры C результат прямой суммы $A \oplus C$ (выигрыш или проигрыш) такой же, как результат $B \oplus C$.

Можно так же определить для игр с любым количеством фишек.

Свойство эквивалентности

Полезное свойство эквивалентности:

Свойство эквивалентности

Полезное свойство эквивалентности:

Утверждение 3: Игры A и B, каждая с одной фишкой, эквивалентны $\Leftrightarrow$ числа Гранди под фишками в них равны.

Свойство эквивалентности

Полезное свойство эквивалентности:

Утверждение 3: Игры A и B, каждая с одной фишкой, эквивалентны $\Leftrightarrow$ числа Гранди под фишками в них равны.

- если $g(A) \neq g(B)$, рассмотрим $C = A$:

Свойство эквивалентности

Полезное свойство эквивалентности:

Утверждение 3: Игры A и B, каждая с одной фишкой, эквивалентны $\Leftrightarrow$ числа Гранди под фишками в них равны.

- если $g(A) \neq g(B)$, рассмотрим $C = A$:
- $A \oplus A$ проигрышная, $B \oplus A$ выигрышная

Свойство эквивалентности

Полезное свойство эквивалентности:

Утверждение 3: Игры A и B, каждая с одной фишкой, эквивалентны $\Leftrightarrow$ числа Гранди под фишками в них равны.

- если $g(A) \neq g(B)$, рассмотрим $C = A$:
- $A \oplus A$ проигрышная, $B \oplus A$ выигрышная
- если $g(A) = g(B)$, то для любой игры C:

Свойство эквивалентности

Полезное свойство эквивалентности:

Утверждение 3: Игры A и B, каждая с одной фишкой, эквивалентны $\Leftrightarrow$ числа Гранди под фишками в них равны.

- если $g(A) \neq g(B)$, рассмотрим $C = A$:
- $A \oplus A$ проигрышная, $B \oplus A$ выигрышная
- если $g(A) = g(B)$, то для любой игры C:
- либо $g(C) = g(A)$ и $g(C) = g(B)$,

Свойство эквивалентности

Полезное свойство эквивалентности:

Утверждение 3: Игры A и B, каждая с одной фишкой, эквивалентны $\Leftrightarrow$ числа Гранди под фишками в них равны.

- если $g(A) \neq g(B)$, рассмотрим $C = A$:
- $A \oplus A$ проигрышная, $B \oplus A$ выигрышная
- если $g(A) = g(B)$, то для любой игры C:
- либо $g(C) = g(A)$ и $g(C) = g(B)$,
- либо $g(C) \neq g(A)$ и $g(C) \neq g(B)$

План решения задач

Мы хотим научиться узнавать для прямой суммы нескольких сложных игр, каким будет результат при правильной игре.

План решения задач

Мы хотим научиться узнавать для прямой суммы нескольких сложных игр, каким будет результат при правильной игре.

- сначала заменим сложные игры на простые:

План решения задач

Мы хотим научиться узнавать для прямой суммы нескольких сложных игр, каким будет результат при правильной игре.

- сначала заменим сложные игры на простые:
- любой игре A с одной фишкой эквивалентен ним с $n = g(A)$

План решения задач

Мы хотим научиться узнавать для прямой суммы нескольких сложных игр, каким будет результат при правильной игре.

- сначала заменим сложные игры на простые:
- любой игре A с одной фишкой эквивалентен ним с $n = g(A)$
- эквивалентность означает, что в прямой сумме игр эти игры взаимозаменяемы

План решения задач

Мы хотим научиться узнавать для прямой суммы нескольких сложных игр, каким будет результат при правильной игре.

- сначала заменим сложные игры на простые:
- любой игре A с одной фишкой эквивалентен ним с $n = g(A)$
- эквивалентность означает, что в прямой сумме игр эти игры взаимозаменяемы
- осталось научиться из нескольких простых игр (нимов) делать одну игру с одной фишкой...

План решения задач

Мы хотим научиться узнавать для прямой суммы нескольких сложных игр, каким будет результат при правильной игре.

- сначала заменим сложные игры на простые:
- любой игре A с одной фишкой эквивалентен ним с $n = g(A)$
- эквивалентность означает, что в прямой сумме игр эти игры взаимозаменяемы
- осталось научиться из нескольких простых игр (нимов) делать одну игру с одной фишкой...
- достаточно научиться из двух нимов делать один ним

Два нима

Как для двух нимов найти игру, эквивалентную их прямой сумме?

Два нима

Как для двух нимов найти игру, эквивалентную их прямой сумме?

- рассмотрим прямую сумму двух нимов

Два нима

Как для двух нимов найти игру, эквивалентную их прямой сумме?

- рассмотрим прямую сумму двух нимов
- пусть в одном число Гранди равно a, а в другом b

Два нима

Как для двух нимов найти игру, эквивалентную их прямой сумме?

- рассмотрим прямую сумму двух нимов
- пусть в одном число Гранди равно a, а в другом b
- с одной стороны, это два отдельных графа с фишкой на каждом

Два нима

Как для двух нимов найти игру, эквивалентную их прямой сумме?

- рассмотрим прямую сумму двух нимов
- пусть в одном число Гранди равно a, а в другом b
- с одной стороны, это два отдельных графа с фишкой на каждом
- с другой стороны, можно нарисовать эту игру как прямоугольную таблицу: из каждой клетки есть ходы строго влево и строго вниз

Два нима

Как для двух нимов найти игру, эквивалентную их прямой сумме?

- рассмотрим прямую сумму двух нимов
- пусть в одном число Гранди равно a, а в другом b
- с одной стороны, это два отдельных графа с фишкой на каждом
- с другой стороны, можно нарисовать эту игру как прямоугольную таблицу: из каждой клетки есть ходы строго влево и строго вниз
- это граф из $(a+1) \times (b+1)$ вершин, зато с одной фишкой

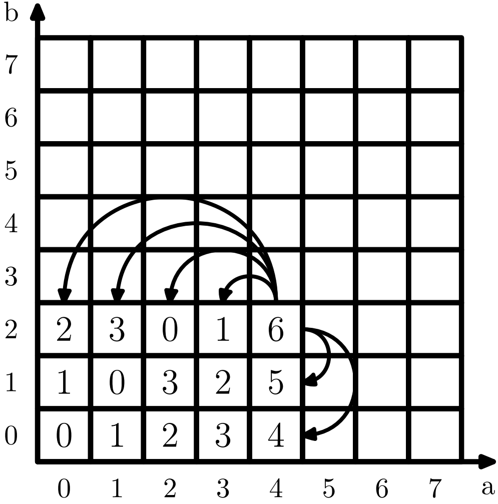
Два нима

Как для двух нимов найти игру, эквивалентную их прямой сумме?

- рассмотрим прямую сумму двух нимов
- пусть в одном число Гранди равно a, а в другом b
- с одной стороны, это два отдельных графа с фишкой на каждом
- с другой стороны, можно нарисовать эту игру как прямоугольную таблицу: из каждой клетки есть ходы строго влево и строго вниз
- это граф из $(a+1) \times (b+1)$ вершин, зато с одной фишкой
- осталось узнать число под этой фишкой

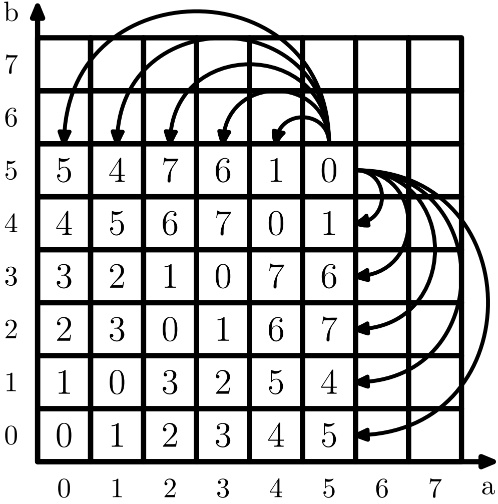
Таблица

Вот эта двумерная таблица:



Таблица

Вот эта двумерная таблица:



b \ a	0	1	2	3	4	5	6	7
7								
6								
5	5	4	7	6	1	0		
4	4	5	6	7	0	1		
3	3	2	1	0	7	6		
2	2	3	0	1	6	7		
1	1	0	3	2	5	4		
0	0	1	2	3	4	5		

Таблица

Вот эта двумерная таблица:

b									
7	7	6	5	4	3	2	1	0	
6	6	7	4	5	2	3	0	1	
5	5	4	7	6	1	0	3	2	
4	4	5	6	7	0	1	2	3	
3	3	2	1	0	7	6	5	4	
2	2	3	0	1	6	7	4	5	
1	1	0	3	2	5	4	7	6	
0	0	1	2	3	4	5	6	7	
	0	1	2	3	4	5	6	7	a

Теорема Шпрага–Гранди

Теорема Шпрага–Гранди (Sprague–Grundy):

Пусть A и B — две игры.

Тогда $g(A \oplus B) = g(A)$ xor $g(B)$.

Теорема Шпрага–Гранди

Теорема Шпрага–Гранди (Sprague–Grundy):

Пусть A и B — две игры.
Тогда $g(A \oplus B) = g(A)$ xor $g(B)$.

- здесь xor — побитовая операция «исключающее или»

Теорема Шпрага–Гранди

Теорема Шпрага–Гранди (Sprague–Grundy):

Пусть A и B — две игры.
Тогда $g(A \oplus B) = g(A)$ xor $g(B)$.

- здесь xor — побитовая операция «исключающее или»
- пример:
- $\mathtt{x} = 5 = \ldots 0101_2$
- $\mathtt{y} = 3 = \ldots 0011_2$
- $\mathtt{z} = 6 = \ldots 0110_2 = \mathtt{x}$ xor $\mathtt{y}$

Иллюстрация

Как строится таблица:

b \ a	0	1	2	3	4	5	6	7
7	7	6	5	4	3	2	1	0
6	6	7	4	5	2	3	0	1
5	5	4	7	6	1	0	3	2
4	4	5	6	7	0	1	2	3
3	3	2	1	0	7	6	5	4
2	2	3	0	1	6	7	4	5
1	1	0	3	2	5	4	7	6
0	0	1	2	3	4	5	6	7

Доказательство теоремы

Докажем теорему:

- будем строить таблицу для прямой суммы индукцией по размеру: 1×1, 2×2, 4×4, 8×8, $\ldots$
- рассмотрим подробно переход $4 \times 4 \longrightarrow 8 \times 8$
- как получается верхний левый квадрат 4×4?
- снизу в каждом столбце есть числа $0, 1, 2, 3$
- значит, можно считать, что вычисление mex просто начинается не с 0, а с 4
- в остальном этот квадрат строится совершенно так же, как и нижний левый
- поэтому он получается из нижнего левого квадрата, если к каждому числу прибавить 4

Доказательство теоремы

Докажем теорему:

- будем строить таблицу для прямой суммы индукцией по размеру: 1×1, 2×2, 4×4, 8×8, $\ldots$
- рассмотрим подробно переход $4 \times 4 \longrightarrow 8 \times 8$
- как получается нижний правый квадрат 4×4?
- слева в каждой строке есть числа $0, 1, 2, 3$
- значит, можно считать, что вычисление mex просто начинается не с 0, а с 4
- в остальном этот квадрат строится совершенно так же, как и нижний левый
- поэтому он получается из нижнего левого квадрата, если к каждому числу прибавить 4

Доказательство теоремы

Докажем теорему:

- будем строить таблицу для прямой суммы индукцией по размеру: 1×1, 2×2, 4×4, 8×8, ...
- рассмотрим подробно переход $4 \times 4 \longrightarrow 8 \times 8$
- как получается верхний правый квадрат 4×4?
- снизу в каждом столбце есть числа $4, 5, 6, 7$
- слева в каждой строке есть числа $4, 5, 6, 7$
- этот квадрат строится совершенно так же, как и нижний левый, и числа 4 и больше не получаются, значит, они не влияют на результат
- поэтому этот квадрат — просто копия нижнего левого квадрата без изменений

Доказательство теоремы

Доказажем теорему:

- будем строить таблицу для xor индукцией по размеру: 1×1, 2×2, 4×4, 8×8, ...
- рассмотрим подробно переход $4 \times 4 \longrightarrow 8 \times 8$
- как получается верхний левый квадрат 4×4?
- копируем нижний левый квадрат и прибавляем к каждому числу 4
- как получается нижний правый квадрат 4×4?
- копируем нижний левый квадрат и прибавляем к каждому числу 4
- как получается верхний правый квадрат 4×4?
- копируем нижний левый квадрат без изменений
- итак, две таблицы равны по построению, что и требовалось доказать

Решение исходной задачи

Что, если скомбинировать три игры?

- три кучки: a, b и c камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?

Решение исходной задачи

Что, если скомбинировать три игры?

- три кучки: a, b и c камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?
- $a \text{ xor } b \text{ xor } c = 0 \Leftrightarrow$ победит второй игрок

Решение исходной задачи

Что, если скомбинировать три игры?

- три кучки: a, b и c камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?
- a xor b xor $c = 0 \Leftrightarrow$ победит второй игрок
- пример $a = 1$, $b = 2$, $c = 3$: a xor b xor $c = 1$ xor 2 xor 3 $= 3$ xor $3 = 0$, значит, победит второй игрок

Решение исходной задачи

Что, если скомбинировать три игры?

- три кучки: a, b и c камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?
- a xor b xor $c = 0 \Leftrightarrow$ победит второй игрок
- пример $a = 1$, $b = 2$, $c = 3$: a xor b xor $c = 1$ xor 2 xor 3 $= 3$ xor $3 = 0$, значит, победит второй игрок
- пример $a = 24$, $b = 47$, $c = 55$:
  a xor b xor $c = 24$ xor 47 xor 55
  $= 55$ xor $55 = 0$, значит, победит второй игрок

Решение исходной задачи

Что, если скомбинировать три игры?

- три кучки: a, b и c камней
- за ход можно взять любое положительное число камней, но только из одной кучки
- кто победит?
- $a \text{ xor } b \text{ xor } c = 0 \Leftrightarrow$ победит второй игрок
- пример $a = 1$, $b = 2$, $c = 3$: $a \text{ xor } b \text{ xor } c = 1 \text{ xor } 2 \text{ xor } 3$ $= 3 \text{ xor } 3 = 0$, значит, победит второй игрок
- пример $a = 24$, $b = 47$, $c = 55$:
  $a \text{ xor } b \text{ xor } c = 24 \text{ xor } 47 \text{ xor } 55$
  $= 55 \text{ xor } 55 = 0$, значит, победит второй игрок
- пример $a = 23$, $b = 48$, $c = 55$:
  $a \text{ xor } b \text{ xor } c = 23 \text{ xor } 48 \text{ xor } 55$
  $= 39 \text{ xor } 55 = 16$, значит, победит первый игрок

Выбор хода

Предположим, что позиция выигрышная.
Как выбирать правильный ход?

- самое простое:
- перебрать все возможные ходы

Выбор хода

Предположим, что позиция выигрышная.
Как выбирать правильный ход?

- самое простое:
- перебрать все возможные ходы
- пересчитать число Гранди после каждого возможного хода

Выбор хода

Предположим, что позиция выигрышная.
Как выбирать правильный ход?

- самое простое:
- перебрать все возможные ходы
- пересчитать число Гранди после каждого возможного хода
- выбрать тот ход, после которого оно равно нулю

Выбор хода

Предположим, что позиция выигрышная.
Как выбирать правильный ход?

- самое простое:
- перебрать все возможные ходы
- пересчитать число Гранди после каждого возможного хода
- выбрать тот ход, после которого оно равно нулю
- это и будет ход в проигрышную (для противника) позицию

Пример

Вот ещё один пример.

- n кучек камней: $a_1, a_2, \ldots, a_n$ камней
- за ход можно взять 2, 3 или 5 камней, но только из одной кучки
- кто победит?

Пример

Вот ещё один пример.

- n кучек камней: $a_1, a_2, \ldots, a_n$ камней
- за ход можно взять 2, 3 или 5 камней, но только из одной кучки
- кто победит?
- для каждого возможного количества камней k в кучке заранее посчитаем число Гранди $g(k)$

Пример

Вот ещё один пример.

- n кучек камней: $a_1, a_2, \dots, a_n$ камней
- за ход можно взять 2, 3 или 5 камней, но только из одной кучки
- кто победит?
- для каждого возможного количества камней k в кучке заранее посчитаем число Гранди $g(k)$
- $g(k) = \operatorname{mex}\{g(k-2), g(k-3), g(k-5)\}$

Пример

Вот ещё один пример.

- n кучек камней: $a_1, a_2, \ldots, a_n$ камней
- за ход можно взять 2, 3 или 5 камней, но только из одной кучки
- кто победит?
- для каждого возможного количества камней k в кучке заранее посчитаем число Гранди $g(k)$
- $g(k) = \text{mex}\{g(k-2), g(k-3), g(k-5)\}$
- перед вычислением и добавлением каждого $g(k-x)$ проверяем, что $k-x$ неотрицательно

Пример

Вот ещё один пример.

- n кучек камней: $a_1, a_2, \ldots, a_n$ камней
- за ход можно взять 2, 3 или 5 камней, но только из одной кучки
- кто победит?
- для каждого возможного количества камней k в кучке заранее посчитаем число Гранди $g(k)$
- $g(k) = \mathrm{mex}\{g(k-2), g(k-3), g(k-5)\}$
- перед вычислением и добавлением каждого $g(k-x)$ проверяем, что $k-x$ неотрицательно
- наконец, вычислим $g(a_1)$ xor $g(a_2)$ xor $\ldots$ xor $g(a_n)$

Пример

Вот ещё один пример.

- n кучек камней: $a_1, a_2, \ldots, a_n$ камней
- за ход можно взять 2, 3 или 5 камней, но только из одной кучки
- кто победит?
- для каждого возможного количества камней k в кучке заранее посчитаем число Гранди $g(k)$
- $g(k) = \operatorname{mex}\{g(k-2), g(k-3), g(k-5)\}$
- перед вычислением и добавлением каждого $g(k-x)$ проверяем, что $k-x$ неотрицательно
- наконец, вычислим $g(a_1)$ xor $g(a_2)$ xor $\ldots$ xor $g(a_n)$
- если получился 0, текущая позиция проигрышная

Пример

Вот ещё один пример.

- n кучек камней: $a_1, a_2, \ldots, a_n$ камней
- за ход можно взять 2, 3 или 5 камней, но только из одной кучки
- кто победит?
- для каждого возможного количества камней k в кучке заранее посчитаем число Гранди $g(k)$
- $g(k) = \mathrm{mex}\{g(k-2), g(k-3), g(k-5)\}$
- перед вычислением и добавлением каждого $g(k-x)$ проверяем, что $k-x$ неотрицательно
- наконец, вычислим $g(a_1)$ xor $g(a_2)$ xor $\ldots$ xor $g(a_n)$
- если получился 0, текущая позиция проигрышная
- если получился не 0, текущая позиция выигрышная

Код

Код для предыдущего примера.

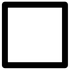
```
#include <iostream>
#include <set>
#include <vector>
using namespace std;
vector <int> moves = {2, 3, 5};
int const limit = 100;
int main () {
    vector <int> g (limit + 1);
    for (int v = 0; v <= limit; v++) {
        set <int> s;
        for (auto k : moves) if (v >= k) s.insert (g[v - k]);
        g[v] = 0; while (s.count (g[v])) g[v] += 1;
    }
    int n; cin >> n;
    vector <int> a (n);
    int total = 0;
    for (int i = 0; i < n; i++) {
        cin >> a[i]; total ^= g[a[i]];
    }
    cout << (total == 0 ? "Second" : "First") << endl;
    return 0;
}
```

Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней

Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



0

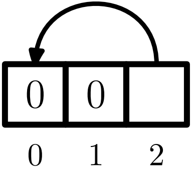
Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней

0	
0	1

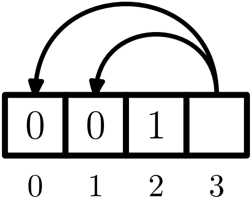
Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



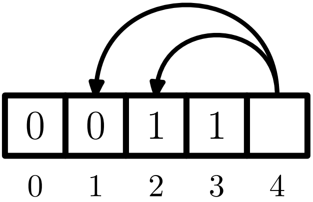
Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



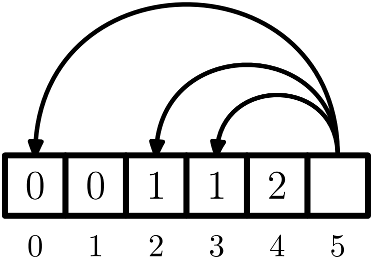
Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



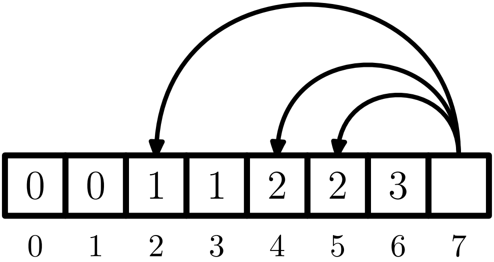
Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней

0	0	1	1	2	2	
0	1	2	3	4	5	6

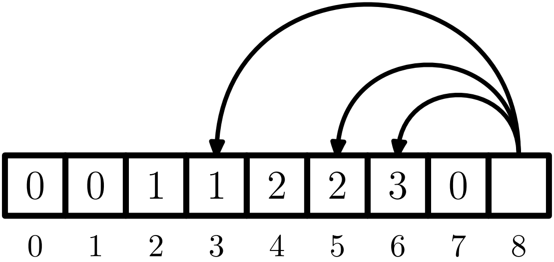
Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



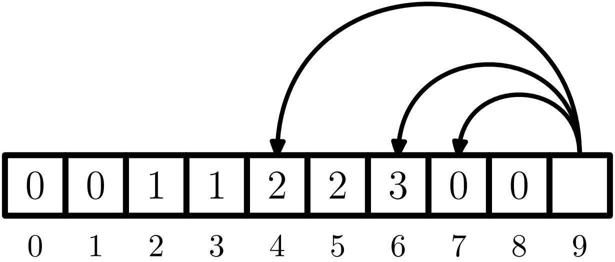
Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



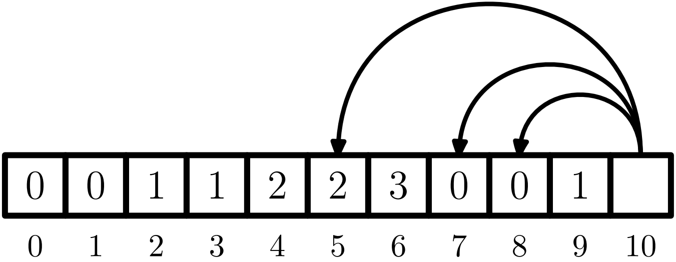
Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



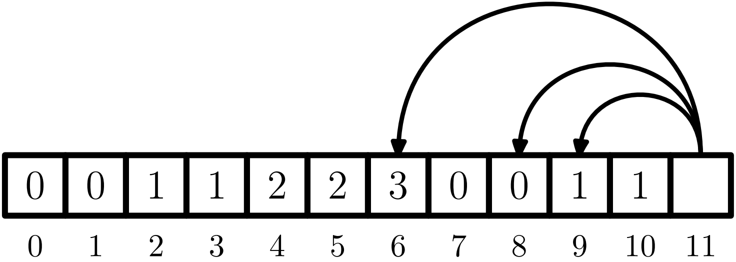
Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



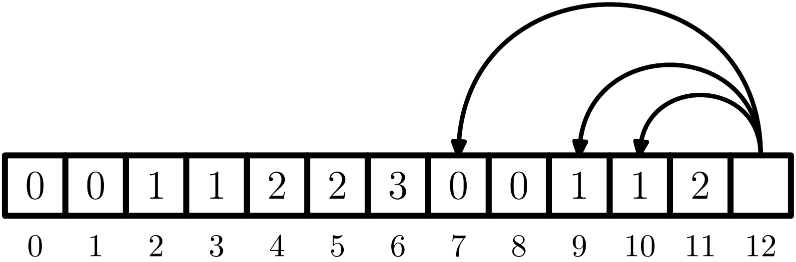
Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



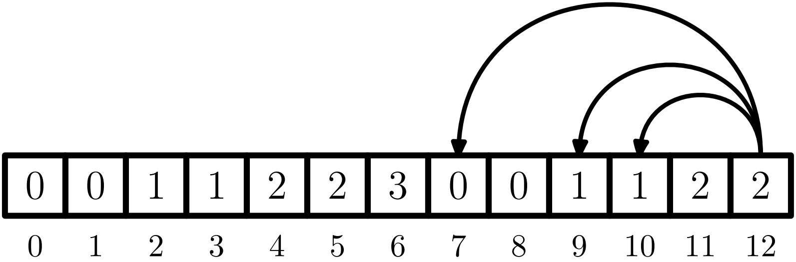
Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



Таблица

Числа Гранди: за ход можно взять 2, 3 или 5 камней



Вопросы?

Вопросы?