

## Теория игр. Задачи 19-21.

Функция для реализации игры с камнями на Python

```
def f(x, m): # x - это количество камней в куче, m - кол-во ходов до окончания игры

    if x <= 19: #условие победы
        return m%2 == 0 #проверка того, кто из двух игроков победил

    if m == 0: return 0 #если условие победы не достигнуто за выделенное число ходов, то игра завершается

    games = [f(x-2, m-1), f(x-5, m-1), f(x//3, m-1)] # список со всеми вариантами ходов
    return any(games) if m%2 != 0 else all(games) # продолжение игры, если условие победы не достигнуто, а лимит ходов не исчерпан
```

Наличие выигрышной стратегии означает, что на все варианты хода игрока, который должен проиграть, у игрока, обладающего этой стратегией, найдется  $\geq 1$  ход, ведущий к победе, поэтому используются проверки `any(games)` и `all(games)`

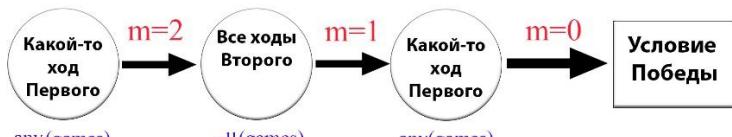
Визуализация алгоритма для  $m = 4$  и  $m = 3$

$m=4$ , где  $m$  - количество ходов до окончания игры



если игра заканчивается за нечетное число ходов, то последний победный ход делает тот, кто начинал первый, а если за четное - второй.

$m=3$ , где  $m$  - количество ходов до окончания игры



если игра заканчивается за нечетное число ходов, то последний победный ход делает тот, кто начинал первый, а если за четное - второй.

Таким образом, победа первого игрока равносильна тому, что игра закончится за нечетное число ходов, а победа второго – тому, что игра закончится за четное число ходов.

Причем тот, кто должен победить, своим последним ходом обнуляет значение  $m$ . Вследствие этого значение  $m$  на ходах этого игрока всегда четно: 0 – на его последнем ходе, 2 на его предпоследнем ходе и т.д.

Поэтому используется проверка `m%2 == 0`, а не `m == 0` для учёта случаев, когда игра завершается раньше обнуления  $m$ , но на ходу того же игрока, который должен победить по условию.

Это объясняет и строчку `return any(h) if m%2 != 0 else all(h)`

Если  $m$  – четно, то ход выполнял тот, кто должен победить  $\Rightarrow$  следующий ход делает тот, кто должен проиграть, и мы должны проверить все его ходы (это видно на рисунке).

**19.** Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может: убрать из кучи два камня или убрать из кучи пять камней или уменьшить количество камней в куче в три раза (количество камней, полученное при делении, округляется до меньшего). Например, из кучи в 20 камней за один ход можно получить кучу из 18, 15 или 6 камней. Игра завершается, когда количество камней в куче становится не более 19.

Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 19 или меньше камней. В начальный момент в куче было  $S$  камней,  $S \geq 20$ .

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Укажите минимальное значение  $S$ , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.

**20.** Для игры, описанной в задании 19, найдите два наименьших значения  $S$ , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Найденные значения запишите в ответе в порядке возрастания.

**21.** Для игры, описанной в задании 19, найдите минимальное значение  $S$ , при котором одновременно выполняются два условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

**19.** Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу от 1 до 3 камней либо увеличить количество камней в куче в 2 раза. Для того, чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 172. Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, состоящую из 172 или более камней. В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 171$ . Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Укажите такое значение  $S$ , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.

**20.** Для игры, описанной в задании 19, найдите наименьшее и наибольшее значения  $S$ , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Найденные значения запишите в ответе в порядке возрастания.

**21.** Для игры, описанной в задании 19, найдите такое значение  $S$ , при котором одновременно выполняются два

условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Если найдено несколько значений  $S$ , в ответе запишите наименьшее из них.

## Ответы и решение.

[№ 17875 Демоверсия 2025 (Уровень: Базовый)]

```
def f(x, m):
    if x <= 19: return m%2 == 0
    if m == 0: return 0
    h = [f(x-2, m-1), f(x-5, m-1), f(x//3, m-1)]
    return any(h) if m%2 != 0 else all(h)
```

19. Ответ: 60

```
for s in range(20, 1000):
    if not f(s, 1) and f(s, 2): print(s)
```

20. Ответ: 62 63

```
for s in range(20, 1000):
    if not f(s, 1) and f(s, 3): print(s)
```

21. Ответ: 64

```
for s in range(20, 1000):
    if not f(s, 2) and f(s, 4): print(s)
```

[№ 13961 (Уровень: Базовый)]

```
def f(x, m):
    if x >= 172: return m % 2 == 0
    if m == 0: return 0
    h = [f(x+i, m-1) for i in (1, 2, 3)] + [f(x*2, m-1)]
    return any(h) if m%2 != 0 else all(h)
```

19. Ответ: 85

```
for s in range(1, 172):
    if not f(s, 1) and f(s, 2): print(s)
```

20. Ответ: 82 84

```
for s in range(20, 1000):
    if not f(s, 1) and f(s, 3): print(s)
```

20. Ответ: 81

```
for s in range(20, 1000):
    if not f(s, 2) and f(s, 4): print(s)
```