

Информатика. Программирование. Теория игр. Неудачный ход.

Содержание

1	Разбор задач с вебинара	2
1.1	Пример 1	2
1.2	Пример 2	3
1.3	Пример 3	4
1.4	Пример 4	5
1.5	Пример 5	6
1.6	Пример 6	7
1.7	Пример 7	8
1.8	Пример 8	10

1 Разбор задач с вебинара

1.1 Пример 1

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 1 камень или добавить в кучу 10 камней. Например, имея кучу из 7 камней, за один ход можно получить кучу из 8 или 17 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 61. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 61 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 55$.

Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите минимальное значение S , когда такая ситуация возможна.

Решение:

```
def f(a):
    if a >= 61:
        return 0
    t = [f(a+1), f(a+10)]
    n = [i for i in t if i <= 0]
    if n:
        return -max(n)+1
    else:
        return -max(t)

# Список i, из которых Петя может сделать выигрышную позицию для Вани
t = [i for i in range(1,56) if f(i+1) == 1 or f(i+10) == 1]
print(min(t)) # Минимальное значение i
```

Ответ: 41

1.2 Пример 2

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 один или три камня или увеличить количество камней в куче в 2 раза, после каждого хода в куче должно быть четное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 88.

Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в куче будет 88 или более камней. В начальный момент в куче было S камней, $1 \leq S \leq 87$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Найдите минимальное значение S , при котором Ваня выигрывает своим первым ходом при любой игре Пети.

Решение:

```
def f(a):
    if a >= 88:
        return 0
    t = [f(a*2)] # Ход, доступный для любого количества камней
    # Ходы +1 и +3 только для нечетного количества камней,
    # чтобы получить четное количество камней
    if a % 2 != 0:
        t += [f(a+1), f(a+3)]
    n = [i for i in t if i <= 0]
    if n:
        return -max(n)+1
    else:
        return -max(t)
for i in range(1,88):
    if f(i) == -1:
        print(i)
```

Ответ: 22

1.3 Пример 3

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 один или два камня** или **увеличить количество камней в куче в два раза**. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 29. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 29 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 28$.

Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите минимальное значение S , когда такая ситуация возможна.

Решение:

```
from functools import*
@lru_cache(None)
def f(a):
    if a >= 29:
        return 0
    t = [f(a+1), f(a+2), f(a*2)]
    n = [i for i in t if i <= 0]
    if n:
        return -max(n)+1
    else:
        return -max(t)

for i in range(1,56):
    # Значения i, из которых Петя может сделать выигрышную позицию для Вани
    if f(i+1) == 1 or f(i+2) == 1 or f(i*2) == 1:
        print(i)
```

Ответ: 8

1.4 Пример 4

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить в одну из куч два камня** или **увеличить количество камней в куче в два раза**. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в кучах становится не менее 40.

В начальный момент в первой куче было 8 камней, во второй куче — S камней; $1 \leq S \leq 31$. Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите минимальное значение S , когда такая ситуация возможна.

Решение:

```
from functools import*
@lru_cache(None)
def f(a,b):
    if a + b >= 40:
        return 0
    t = [f(a+2,b), f(a*2,b), f(a,b+2), f(a,b*2)]
    n = [i for i in t if i <= 0]
    if n:
        return -max(n)+1
    else:
        return -max(t)

for i in range(1,31):
    # Значения i, из которых Петя может сделать выигрышную позицию для Вани
    if f(8+2,i) == 1 or f(8*2,i) == 1 or f(8,i+2) == 1 or f(8,i*2) == 1:
        print(i)
```

Ответ: 8

1.5 Пример 5

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч два камня или увеличить количество камней в куче в два раза. Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 79. Победителем считается игрок, сделавший последний ход, т. е. первым получивший позицию, в которой в кучах будет 79 или больше камней. В начальный момент в первой куче было 9 камней, во второй куче – S камней, $1 \leq S \leq 69$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Назовите минимальное значение S , при котором это возможно.

Решение:

```
from functools import*
@lru_cache(None)
def f(a,b):
    if a + b >= 79:
        return 0
    t = [f(a+2,b), f(a*2,b), f(a,b+2), f(a,b*2)]
    n = [i for i in t if i <= 0]
    if n:
        return -max(n)+1
    else:
        return -max(t)
for i in range(1,70):
    # Значения i, из которых Петя может сделать выигрышную позицию для Вани
    if f(9+2,i) == 1 or f(9*2,i) == 1 or f(9,i+2) == 1 or f(9,i*2) == 1:
        print(i)
```

Ответ: 18

1.6 Пример 6

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить в одну из куч четыре камня** или **увеличить количество камней в куче в два раза**. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 74. Победителем считается игрок, сделавший последний ход, т. е. первым получивший такую позицию, при которой в кучах будет 74 или больше камней.

В начальный момент в первой куче было 10 камней, во второй куче — S камней; $1 \leq S \leq 63$. Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите минимальное значение S , когда такая ситуация возможна

Решение:

```
from functools import*
@lru_cache(None)
def f(a,b):
    if a + b >= 74:
        return 0
    t = [f(a+4,b), f(a*2,b), f(a,b+4), f(a,b*2)]
    n = [i for i in t if i <= 0]
    if n:
        return -max(n)+1
    else:
        return -max(t)
for i in range(1,63):
    # Значения i, из которых Петя может сделать выигрышную позицию для Вани
    if f(10+4,i) == 1 or f(10*2,i) == 1 or f(10,i+4) == 1 or f(10,i*2) == 1:
        print(i)
```

Ответ: 16

1.7 Пример 7

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может убрать в одной из куч два камня или уменьшить количество камней в куче в два раза (целочисленное деление). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Количество камней не может быть отрицательно и игрок не может производить действие с кучей если она пуста. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не более 12. Победителем считается игрок, сделавший последний ход, т. е. первым получивший позицию, в которой в кучах будет 12 или меньше камней.

В начальный момент в первой куче было 25 камней, во второй куче — S камней, $1 \leq S \leq 99$. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Назовите минимальное значение S , при котором это возможно.

Решение:

```
from functools import*
@lru_cache(None)
def f(a,b):
    if a + b <= 12:
        return 0
    # Проверка, чтобы не получить отрицательное значение камней
    # и не превысить глубину рекурсии ходом // 2
    if a == 0:
        t = [f(a,b-2), f(a,b//2)]
    elif b == 0:
        t = [f(a-2,b),f(a//2,b)]
    # Проверка, чтобы не получить отрицательное значение камней ходом -2
    elif a == 1:
        t = [f(a,b-2), f(a,b//2), f(a//2,b)]
    elif b == 1:
        t = [f(a-2,b),f(a//2,b), f(a,b//2)]
    else:
        t = [f(a-2,b), f(a//2,b), f(a,b-2), f(a,b//2)]
    n = [i for i in t if i <= 0]
    if n:
        return -max(n)+1
    else:
        return -max(t)

for i in range(1,100):
    # Значения i, из которых Петя может сделать выигрышную позицию для Вани
    if f(25-2,i) == 1 or f(25//2,i) == 1 or f(25,i-2) == 1 or f(25,i//2) == 1:
        print(i)
```

Ответ: 1

1.8 Пример 8

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может увеличить количество камней во второй куче в два раза или поменять кучи местами. Второй раз менять кучи, если это уже сделал противник, нельзя. Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 199. Победителем считается игрок, сделавший последний ход, т. е. первым получивший позицию, в которой в кучах будет 199 или больше камней.

В начальный момент в первой куче было 99 камней, во второй куче — S камней, $1 \leq S \leq 99$. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Назовите минимальное значение S , при котором это возможно.

Решение:

```
from functools import*
@lru_cache(None)
# move - номер команды, сделанной на предыдущем ходе
def f(a,b, move):
    if a + b >= 199:
        return 0
    if move != 2: # Если предыдущая команда была не 2,
        t = [f(a,b*2,1),f(b,a,2)] # можно делать оба хода
    else: # Иначе только первый ход
        t = [f(a,b*2,1)]
    n = [i for i in t if i <= 0]
    if n:
        return -max(n)+1
    else:
        return -max(t)
for i in range(1,100):
    # Значения i, из которых Петя может сделать выигрышную позицию для Вани
    if f(99,i*2,1) == 1 or f(i,99,2) == 1:
        print(i)
```

Ответ: 1