

# Конспект по заданиям №19-21

## ТЕОРИЯ ИГР

### 1 ПРИМЕР (ОДНА КУЧА КАМНЕЙ)

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 один или четыре камня либо увеличить количество камней в куче в три 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 67.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, состоящую из 67 или более камней.

В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 66$ .

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.

```
from functools import lru_cache
def moves(x):
    return (x+1),(x+4),(x*3)
@lru_cache(None)
def f(x):
    if x>=67:
        return "win"
    if any(f(i)=="win" for i in moves(x)):
        return "p1"
    if all(f(i)=="p1" for i in moves(x)): # МЕНЯЕМ на all (при любом ходе)
        return "w1"
    for s in range(1,67):
        if f(s)=="w1":
            print(s)
```

```
from functools import lru_cache
def moves(x):
    return (x+1),(x+4),(x*3)
```

```

@lru_cache(None)
def f(x):
    if x>=67:
        return "win"
    if any(f(i)=="win" for i in moves(x)):
        return "p1"
    if all(f(i)=="p1" for i in moves(x)):
        return "w1"
    if any(f(i)=="w1" for i in moves(x)):
        return "p2"
    for s in range(1,67):
        if f(s)=="p2":
            print(s)

```

@hasyanov\_EGE

21

```

from functools import lru_cache
def moves(x):
    return (x+1),(x+4),(x*3)
@lru_cache(None)
def f(x):
    if x>=67:
        return "win"
    if any(f(i)=="win" for i in moves(x)):
        return "p1"
    if all(f(i)=="p1" for i in moves(x)):
        return "w1"
    if any(f(i)=="w1" for i in moves(x)):
        return "p2"
    if all(f(i)=="p1" or f(i)=="p2" for i in moves(x)):
        return "w2"
    for s in range(1,67):
        if f(s)=="w2":
            print(s)

```

## 2 ПРИМЕР (ДВЕ КУЧИ КАМНЕЙ)

19 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 (по своему выбору) один камень 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 81.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, при которой в кучах находится 81 камень или больше. @hasyanov\_EGE

В начальный момент в первой куче было семь камней, во второй куче –  $S$  камней;  $1 \leq S \leq 73$ .

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. Укажите минимальное значение  $S$ , когда такая ситуация возможна.

19

```
from functools import lru_cache
def moves(x,y):
    return (x+1,y),(x,y+1),(x*2,y),(x,y*2)
@lru_cache(None)
def f(x,y):
    if x+y>=81:
        return "win"
    if any(f(*i)=="win" for i in moves(x,y)):
        return "p1"
    if any(f(*i)=="p1" for i in moves(x,y)): #НЕУДАЧНЫЙ ХОД, поэтому апну
        return "w1"
for s in range(1,74):
    if f(7,s)=="w1":
        print(s)
```

20 (снова меняем на all)

```
from functools import lru_cache
def moves(x,y):
    return (x+1,y),(x,y+1),(x*2,y),(x,y*2)
@lru_cache(None)
def f(x,y):
    if x+y>=81:
        return "win"
    if any(f(*i)=="win" for i in moves(x,y)):
        return "p1"
    if all(f(*i)=="p1" for i in moves(x,y)):
        return "w1"
    if any(f(*i)=="w1" for i in moves(x,y)):
        return "p2"
```

```
for s in range(1,74):
    if f(7,s)=="p2":
        print(s)
```

21

```
from functools import lru_cache
def moves(x,y):
    return (x+1,y),(x,y+1),(x*2,y),(x,y*2)
@lru_cache(None)
def f(x,y):
    if x+y>=81:
        return "win"
    if any(f(*i)=="win" for i in moves(x,y)):
        return "p1"
    if all(f(*i)=="p1" for i in moves(x,y)):
        return "w1"
    if any(f(*i)=="w1" for i in moves(x,y)):
        return "p2"
    if all(f(*i)=="p1" or f(*i)=="p2" for i in moves(x,y)):
        return "w2"
for s in range(1,74):
    if f(7,s)=="w2":
        print(s)
```

19

20 (всегда так)

21 (всегда так)

любой ход

any

any

any

all

all

all

any

any

неудачный ход

any

any

all

any