

Задание 19 (Демо 2022) 1 куча

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 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 количество камней в куче становится не менее 29. Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 29 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 28$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии не следует включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т.е. не являющиеся выигрышными независимо от игры 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.

Решение:

Здесь вопрос отличается от прошлой 19-ой задачи. Здесь Петя должен выиграть в любом случае. Мы эту задачу можем воспринимать, как 20-ую из демоверсии 2021. Ведь там тоже игроку нужно обязательно было побеждать. Осталось написать шаблон с соответствующими параметрами.

```
def F(x, p):
    if x >= 29 and p == 3: return True
    if x < 29 and p == 3: return False
    if x >= 29: return False

    if p % 2 == 1:
        return F(x+1, p+1) and F(x*2, p+1)
    else:
        return F(x+1, p+1) or F(x*2, p+1)

for s in range(1, 29):
    if F(s, 1):
        print(s)
```

Заводим функцию F . Т.к. у нас одна куча, то она принимает параметры: x - количество камней в куче, p -позиция игры.

Дальше описываем победу. Если $x \geq 29$ и позиция равна 3 (1 Ход Вани), то возвращаем True, что означает победу.

Если, позиция уже равна 3, но камней меньше, чем должно быть для победы, то возвращаем False (проигрыш).

Третье условие. Если кто-то выиграл, но на первых двух условиях мы не вышли из функции, то, значит, выиграл не тот, кто нам нужен, следовательно, возвращаем False.

Если мы не вышли на первых трёх условиях, то, значит, продолжаем прокручивать ходы, рекурсивно запускаем функцию **F**.

Для нечётных **p** (это ходы Вани), возвращаем разные ходы через **and**, т.к. он должен побеждать в любом случае. При этом увеличиваем на 1 значение **p**.

Для чётных **p** (это ходы Пети), возвращаем ходы через **or**.

В конце перебираем все возможные значения для **s** через цикл **for**, ищем те значения, которые подходят по условию задачи.

Ответ: 14

Задание 20 (Демо 2022)

Для игры, описанной в задании 19, найдите два таких 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.

Решение:

Задача точно такая же, как и в 19 задании, только теперь обязательно должен побежать Петя на своём втором ходу ($p=4$), при любой игре Вани.

Пишем тот же шаблон, немного отредактировав его.

```
def F(x, p):
    if x >= 29 and p == 4: return True
    if x < 29 and p == 4: return False
    if x >= 29: return False

    if p % 2 == 0:
        return F(x+1, p+1) and F(x*2, p+1)
    else:
        return F(x+1, p+1) or F(x*2, p+1)
for s in range(1, 29):
    if F(s, 1):
        print(s)
```

Получается 7 и 13.

Ответ:

7	13
---	----

Задание 21 (Демо 2022) 1 куча

Для игры, описанной в задании 19, найдите 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 минимальное из них.

Решение:

Опять используем прошлый шаблон, но немного модернизируем.

```
def F(x, p):
    if x >= 29 and (p == 3 or p == 5): return True
    if x < 29 and p == 5: return False
    if x >= 29: return False

    if p % 2 == 1:
        return F(x+1, p+1) and F(x*2, p+1)
    else:
        return F(x+1, p+1) or F(x*2, p+1)
```

```
def F1(x, p):
    if x >= 29 and p == 3: return True
    if x < 29 and p == 3: return False
    if x >= 29: return False

    if p % 2 == 1:
        return F1(x+1, p+1) and F1(x*2, p+1)
    else:
        return F1(x+1, p+1) or F1(x*2, p+1)
```

```
for s in range(1, 29):
    if F(s, 1):
        print(s)
print()
```

```
for s in range(1, 29):
    if F1(s, 1):
        print(s)
```

Здесь Ваня должен выигрывать либо на первом своём ходе ($p=3$), либо на втором своём ходе ($p=5$).

Т.к. Ваня не должен гарантированно выиграть своим первым ходом, то мы создаём ещё одну функцию **F1**, похожую на основную функцию **F**, которая вычисляет, когда Ваня именно гарантированно выигрывает на своём первом ходе ($p=3$). И, затем, мы из тех чисел, которые получились в первой функции **F**, исключаем числа, которые получились во второй функции **F1**.

В первой функции получилось 12,14, а во второй 14. Получается ответ 12.

Ответ: 12