

Strim_19_20_21_прога_2

Задача № 1 (2426)

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 увеличить количество камней в куче в три раза, добавить в кучу один камень, или 3 камня, при этом после каждого хода в куче должно быть нечетное количество камней. Например, пусть в куче было 8 камней. Тогда за один ход можно получить кучу из 9 камней или из 11 камней (увеличить количество камней в три раза нельзя, т.к. после этого хода получится четное количество камней – 24). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Выигрывает тот игрок, после хода которого количество камней в куче становится не менее 51.

В начальный момент в куче было S камней; $1 \leq S \leq 50$. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Ответьте на следующие вопросы:

Вопрос 19. Известно, что Ваня выиграл своим первым ходом после первого хода Пети. Назовите минимальное значение S , при котором это возможно.

Вопрос 20. Найдите два наибольших значения S , при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Найденные значения запишите в ответе в порядке возрастания.

Вопрос 21. Сколько существует значений S , при которых у Вани есть выигрышная стратегия, позволяющая ему выиграть вторым ходом при любой игре Пети.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241304?t=0h2m50s

Решение

Напишем функцию, которая будет определять стратегию игроков.

Пропишем ходы. Здесь нужно понимать, что есть два случая.

1) Для четного количества камней, мы можем выполнять ходы:

$$s + 1$$

$$s + 3$$

Ход умножения делать нельзя.

2) Если число будет нечётное, то возможен ход

$$s * 3.$$

Т.е. у нас есть два варианта, в зависимости от того, какое в куче количество камней. В остальном функция не отличается от той, что была подробно рассмотрена в прошлом занятии.

Последовательно проанализируем вопросы задачи.

Первый вопрос. Известно, что Ваня выиграл своим первым ходом после первого хода Пети.

Требуется назвать минимальное значение, при котором это возможно. Заметим, что не указано, что выигрыш должен быть при любом ходе Пети. Т.е., мы можем сказать, что должен быть совершен хотя бы один ход. Следовательно, именно для этого задания, в условии того, что нам возвращает

наша функция заменим «all» на «any». В результате работы программы получаем большой список чисел.

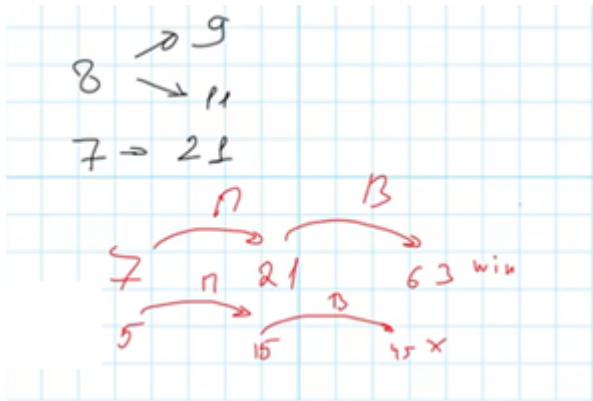
```
def f(s,m):
    if s>=51: return m%2==0
    if m==0: return 0
    if s%2==0:
        h = [f(s+1,m-1),f(s+3,m-1)]
    else:
        h = [f(s*3,m-1)]
    return any(h) if (m-1)%2==0 else any(h)
print(19, [s for s in range(1,51) if f(s,2)])
```

Результат работы программы:

19 [7, 9, 11, 13, 14, 15, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48]

Видим, что минимальное число в этом списке – 7.

Проверим это значение .



Очевидно, что из 7, Петя может пойти только в 21, и, из 21 потом Ваня умножает тоже на 3, получается 63. Если мы возьмем какое-то число поменьше, например, то Петя своим ходом умножает его на 3, получается 15. И, если Ваня потом умножит его на 3, получится 45. Этого значения не хватает для победы. Поэтому, действительно, 7 в данном случае – это минимальное значение.

Второй вопрос. Найдите два наибольших значения S при которых у Пети есть выигрышная стратегия, причем Петя не может выиграть первым ходом, но может выиграть своим вторым ходом, независимо от того, как будет ходить Ваня.

Запишем эти условия в строке для вывода результатов на экран, и заменим обратно условие «any» на «all»

Требуется указать два наибольших числа, поэтому в ответ пойдут числа 12.14.

Вопрос 21. Сколько существует значений S, где Ваня выигрывает своим вторым ходом при любой игре Пети.

Обратим внимание, что если записать это условие, как:

```
print(21, [s for s in range(1,51) if f(s,4)])
```

то в результате будет выведено большое количество чисел, причем большая часть этих чисел – это числа, где Ваня выигрывает только первым ходом, но они все равно попали в это условие. Это

произошло потому, что наша функция ищет победу хотя бы вторым ходом, она ее находит и возвращает такие лишние значения. Поэтому, хотя в условии не прописано, что Ваня не выигрывает гарантированно своим первым ходом, мы это условие добавим, чтобы отсечь эти значения из результата. В итоге у нас останутся только числа 1, 2. Эти два числа, из которых действительно Ваня выиграет своим именно вторым.

```
def f(s,m):
    if s>=51: return m%2==0
    if m==0: return 0
    if s%2==0:
        h = [f(s+1,m-1),f(s+3,m-1)]
    else:
        h = [f(s*3,m-1)]
    return any(h) if (m-1)%2==0 else all(h)
print(20, [s for s in range(1,51) if not f(s,1) and f(s,3)])
print(21, [s for s in range(1,51) if not f(s,2) and f(s,4)])
```

Результат работы программы:

20 [3, 4, 5, 6, 8, 10, 12, 14]

21 [1, 2]

Ответ:

7

12 14

2

Задача №2 (4138)

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит одна куча камней.

Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может

а) убрать из кучи половину камней, если количество камней в куче делится на 2; иначе убрать из кучи два камня;

б) убрать из кучи две трети камней, если количество камней в куче делится на 3; иначе убрать из кучи три камня.

Например, пусть в куче 10 камней, тогда можно убрать половину или только три камня. А если в куче 12 камней, то можно убрать половину или две трети камней. Игра завершается в тот момент, когда в куче останется ровно 1 камень. Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет ровно 1 камень. В начальный момент в куче было S камней, $1 < S \leq 37$.

Ответьте на следующие вопросы:

Вопрос 19. Найдите максимальное значение S , при котором Ваня может выиграть своим первым ходом после неудачного хода Пети, который мог выиграть своим первым ходом.

Вопрос 20. Определите минимальное и максимальное 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.

Вопрос 21. 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.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241304?t=0h21m45s

Решение

Обратим внимание, что каждый ход, в зависимости от ситуации, имеет два различных варианта. В остальном написание функции будет таким же как и во всех ранее разобранных нами задачах.

Список ходов изначально будет пустым.

Далее будем добавлять в него ходы в зависимости от ситуаций указанных в условии задачи:

Если камней четное количество, то в список будет добавляться ход

$s//2$

Иначе, добавляется ход:

$s-2$.

И то же самое касается второго варианта хода:

Если s делится на 3, т.к. по условию игрок убирает из кучи две трети камней, т.е. в ней остается одна треть, в список добавляется ход:

$s//3$,

В противном случае ход:

$s-3$

Вопрос 1. Найдите максимальное значение S , при котором Ваня может выиграть своим первым ходом после неудачного хода Пети, который мог выиграть своим первым ходом.

Т.е. и Петя может выиграть своим первым ходом, но делает неудачный ход, и Ваня выигрывает своим первым ходом. Чтобы учесть это условие мы поменяем `all` на `any`. Запишем условия вывода на экран искомым значений.

Условие

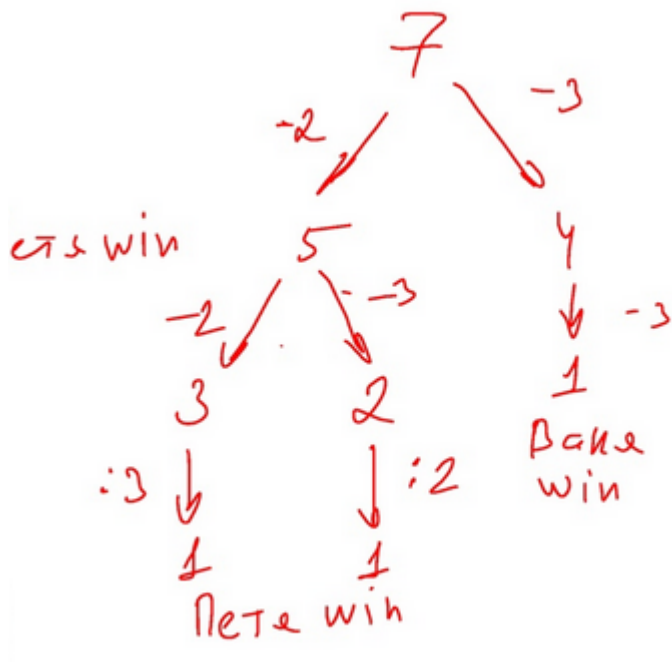
```
if not f(s, 2) and f(s, 4)]])
```

обеспечивает возможность выигрыша и Пети и Вани, если Петин ход будет при неудаче.

Результат работы программы будет представлен единственным числом 4. Это одно число, из которого и Петя может выиграть первым ходом.



Второе задание. В условии вывода на экран учтем что, Петя не может выиграть за один ход, но при любом ходе Ваня выигрывает своим вторым ходом. Из полученных значений проверим минимальный результат:



Очевидно, что результат полученный программой верен.

При выводе значений для ответа на третий вопрос задачи получаем 9, 24, 30, 36 эти числа представляют возможность выигрышной стратегии для Вани вторым ходом.

```
def f(s,m):
    if s==1: return m%2==0
    if m==0: return 0
    h = []
    if s%2==0:
        h.append(f(s//2,m-1))
    else:
        h.append(f(s-2,m-1))
    if s%3==0:
        h.append(f(s//3,m-1))
    else:
        h.append(f(s-3,m-1))
    return any(h) if (m-1)%2==0 else all(h)

#all->any
print(19,[s for s in range(2,38) if f(s,1) and f(s,2)])
print(20,[s for s in range(2,38) if not f(s,1) and f(s,3)])
print(21,[s for s in range(2,38) if not f(s,2) and f(s,4)])
```

Результат работы программы:

```
19 [4]
20 [7, 8, 10, 12, 15, 18]
21 [9, 24, 30, 36]
```

Ответ:

```
4
7 18
9
```

Задача № 3 (6014)

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 изменить количество камней в куче на 1, 3 или 7, но обязательно в сторону числа 42: если в куче больше 42 камней, то он забирает камни из кучи, а если меньше 42, то добавляет камни. Игра завершается в тот момент, когда количество камней в куче становится ровно 42. Игрок, первым получивший кучу из 42 камней, считается победителем.

Ответьте на следующие вопросы:

Вопрос 21. Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите минимальное значение S , когда такая ситуация возможна.

Вопрос 20. 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.

Вопрос 21. Найдите максимальное 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.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241304?t=0h36m20s

Решение

Обратим внимание, как нам следует заполнить список ходов. Если количество камней в куче будет больше 42, то камни из кучи забираются. В противном случае камни следует добавлять. Так как в условии задачи не указан диапазон размеров кучи, будем рассматривать числа от 1 до 100. Для ответа на первый вопрос задачи произведем замену в условии `all->any`

```
def f(s,m):
    if s==42: return m%2==0
    if m==0: return 0
    if s>42:
        h = [f(s-1,m-1), f(s-3,m-1), f(s-7,m-1)]
    else:
        h = [f(s+1,m-1), f(s+3,m-1), f(s+7,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

#all->any
print(19, [s for s in range(1,100) if f(s,2)])
print(20, [s for s in range(1,100) if not f(s,1) and f(s,3)])
print(21, [s for s in range(1,100) if not f(s,2) and f(s,4)])
```

Результат работы программы:

19 [28, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 56]

20 [31, 37, 47, 53]

21 [34, 36, 40, 44, 48, 50]

Из полученных результатов запишем ответ на вопросы задачи. Обратим внимание, что в списке результатов для ответа на первый вопрос число 42 является «импостером», т.е. оно не подходит по условию задачи, т.к. игра сразу может считаться завершённой.

Ответ:

28

31 37

50

Задача № 4 (6292)

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 большую кучу любое количество камней от одного до трёх или удвоить количество камней в меньшей куче. Если кучи содержат равное количество камней, можно добавить в любую из них от одного до трёх камней, удвоение в этой ситуации запрещено.

Игра завершается, когда общее количество камней в двух кучах становится больше или равно 60. Победителем считается игрок, сделавший последний ход, то есть первым получивший 60 или больше камней в двух кучах.

Ответьте на следующие вопросы:

Вопрос 19. Известно, что Петя смог выиграть первым ходом. Какое наименьшее число камней могло быть суммарно в двух кучах?

Вопрос 20. Известно, что в первой куче 12 камней, а во второй – S камней ($1 \leq S \leq 47$). 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.

Вопрос 21. Известно, что в первой куче 25 камней, а во второй – S камней ($1 \leq S \leq 34$). 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

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241304?t=0h44m0s

Решение

Проанализируем условие задачи. Обратим внимание, что в этой задаче две кучи камней и в зависимости от того, какая куча больше или они одинаковые, разные варианты ходов.

Напишем функцию, учитывающую все эти условия. Здесь будет две переменных передающих функции значения соответствующие количествам камней в обеих кучах. Условие выхода из игры будет прописано с учетом наличия этих двух параметров

```
def f(a,b,m):
```

```
if a+b>=60: return m%2==0
if m==0: return 0
```

Распишем все возможные варианты ходов

1) $a > b$, т.е. если a больше b , т.е. первая куча больше? тогда в список ходов попадут ходы увеличивающие первую кучу на 1,2 или 3, или увеличивающие вторую кучу в два раза.

```
h = [f(a+1,b,m-1), f(a+2,b,m-1), f(a+3,b,m-1), f(a,b*2,m-1)]
```

2) $a < b$, то тогда поведение обратное. Мы можем кучу первую только умножить на 2, а прибавлять 1, 2, 3 мы можем только ко второй куче.

```
h = [f(a*2,b,m-1), f(a,b+1,m-1), f(a,b+2,m-1), f(a,b+3,m-1)]
```

3) $a=b$, можно добавлять в любую из куч от одного до трёх камней

```
h = [f(a+1,b,m-1), f(a+2,b,m-1), f(a+3,b,m-1), f(a,b+1,m-1), f(a,b+2,m-1), f(a,b+3,m-1)]
```

Рассмотрим выводы результатов работы этой функции, соответствующие ответам на вопросы задачи
Вопрос 1. Известно, что Петя смог выиграть первым ходом. Какое наименьшее число камней могло быть суммарно в двух кучах? Как вам такой вопрос? Известно, что Петя смог выиграть первым ходом. Какое наименьшее число камней могло быть суммарно в двух кучах?

Для получения ответа на этот вопрос мы будем перебирать пары чисел (x,y) , соответствующие изначальным значениям количеств камней, которые могли быть в кучах. Нужно, понимать, что в паре (x,y) , и это очень важное условие, $(x+y) < 60$, т.к. в противном случае игра закончится, даже не начавшись.

Таким образом мы получим все возможные пары x, y , все возможные комбинации, из первой и второй кучи, если суммарное количество камней в этих кучах было меньше 60. Затем укажем, что из этих начальных количеств Петя может выиграть своим первым ходом.

```
[(x,y) for x in range(1,60) for y in range(1,60) if x+y<60 and f(x,y,1)])
```

С помощью этой строки мы можем получить множество различных пар таких значений, но так как нам требуется получить суммарное значение мы заменим (x,y) на $(x+y)$

И применим к полученному списку функцию `min()`.

Получение ответов на второй и третий вопрос не представляет собой особенных сложностей. Вывод на экран результатов работы функции очевидно соответствуют формулировкам вопросов.

```
def f(a,b,m):
    if a+b>=60: return m%2==0
    if m==0: return 0
    if a>b:
        h = [f(a+1,b,m-1), f(a+2,b,m-1), f(a+3,b,m-1), f(a,b*2,m-1)]
    if a<b:
        h = [f(a*2,b,m-1), f(a,b+1,m-1), f(a,b+2,m-1), f(a,b+3,m-1)]
    if a==b:
        h = [f(a+1,b,m-1), f(a+2,b,m-1), f(a+3,b,m-1), \
            f(a,b+1,m-1), f(a,b+2,m-1), f(a,b+3,m-1)]
    return any(h) if (m-1)%2==0 else all(h)
```

```
print(19,min([x+y for x in range(1,60) for y in range(1,60) if x+y<60 and f(x,y,1)]))
print(20,[s for s in range(1,48) if not f(12,s,1) and f(12,s,3)])
```



```
print(21, [s for s in range(1, 35) if not f(25, s, 2) and f(25, s, 4)])
```

Результат работы программы:

```
19 41
20 [24, 32, 33, 34]
21 [15]
```

Ответ:

```
41
24 34
15
```

Задача №5 (4825)

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, добавить два камня или увеличить количество камней в куче в два раза. При этом нельзя повторять ход, который только что сделал второй игрок. Например, если в начале игры в куче 3 камня, Петя может первым ходом получить кучу из 4, 5 или 6 камней. Если Петя добавил 1 камень и получил кучу из 4 камней, то следующим ходом Ваня может либо добавить 2 камня (и получить 6 камней), либо удвоить количество камней в куче (их станет 8). Получить 5 камней Ваня не может, так как для этого нужно добавить 1 камень, а такой ход только что сделал Петя.

Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается, когда количество камней в куче становится не менее 43. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 43 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 42$.

Ответьте на следующие вопросы:

Вопрос 19. Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.

Вопрос 20. Определите минимальное и максимальное 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.

Вопрос 21. 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.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241304?t=1h0m50s

Решение

Особенность условия этой задачи состоит в том, что нельзя повторять предыдущий ход.

Чтобы контролировать это условие, нам надо, чтобы функция понимала, какой был предыдущий ход. Это можно сделать, добавив специальный параметр. Это будет параметр p , обозначающий значение хода. По нему функция будет определять, какое действие было предыдущим.

```
def f(s, m, p=' '):
```

Обозначим все возможные действия как:

- 1) +1
- 2) +2
- 3) *3

В список `h` будем добавлять возможные ходы в зависимости от того, какой ход не был предыдущим.

```
h = []
if p!='1': h.append(f(s+1,m-1,'1'))
if p!='2': h.append(f(s+2,m-1,'2'))
if p!='3': h.append(f(s*2,m-1,'3'))
```

Т.е. в условии мы указываем, действие, которое было выполнено на предыдущем ходе, чтобы в следующем ходе его не повторять. Соответственно, в список `h` будут добавлены все действия, кроме тех, которые были на предыдущем шаге.

После того, как мы написали функцию, выведем на экран результаты ее работы в соответствии с условиями вопросов. При этом вместо значения предыдущего хода, на месте параметра `p` мы запишем пустые кавычки.

```
print(19,[s for s in range(1,43) if f(s,2.'')])
```

Это сообщить функции о том, что пока вообще не было никакого хода и первый ход может быть любым. В последующих вызовах, этот параметр будет передаваться функции в зависимости от работы условий. Так же мы можем при определении функции прописать значение параметра `p` по умолчанию. Т.е. `p=' '` Это будет означать, если `p` не задано, то оно равно пустые кавычки.

Таким образом в целом код решения задачи будет выглядеть таким образом:

```
def f(s,m,p=' '):
    if s>=43: return m%2==0
    if m==0: return 0
    h = []
    if p!='1': h.append(f(s+1,m-1,'1'))
    if p!='2': h.append(f(s+2,m-1,'2'))
    if p!='3': h.append(f(s*2,m-1,'3'))
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(1,43) if f(s,2)])
print(20,[s for s in range(1,43) if not f(s,1) and f(s,3)])
print(21,[s for s in range(1,43) if not f(s,2) and f(s,4)])
```

Результат работы программы:

```
19 [21]
20 [11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
21 [10]
```

Ответ:

```
21
11 20
10
```

Задание №6 (4375)

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, добавить два камня или увеличить количество камней в куче в два раза. При этом нельзя повторять ход, который этот же игрок делал на предыдущем ходу. Повторять чужие ходы и свои более старые ходы разрешается.

Например, если в начале игры в куче 3 камня, Петя может первым ходом получить кучу из 4, 5 или 6 камней. Если Петя получил кучу из 5 камней (добавил два камня), то следующим ходом Ваня может получить 6, 7 или 10 камней. Если Ваня добавил один камень и получил 6 камней, то вторым ходом Петя может получить 7 или 12 камней. Получить 8 камней Петя не может, так как для этого нужно добавить 2 камня, а Петя делал это на предыдущем ходу.

Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается, 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, $1 \leq S \leq 28$

Задание 19. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Укажите наименьшее значение S , при котором Петя не может выиграть за один ход, но у Пети есть выигрышная стратегия, позволяющая ему выиграть вторым ходом.

Задание 20.

Для игры, описанной в задании 19, укажите два значения S , при которых у Вани есть выигрышная стратегия, позволяющая ему выиграть вторым ходом при любой игре Пети, но у Вани нет стратегии, которая позволяла бы ему гарантированно выиграть первым ходом.

В ответе запишите найденные значения в порядке возрастания: сначала меньшее, затем большее.

Задание 21.

Для игры, описанной в задании 19, найдите наибольшее значение S , при котором у Пети есть выигрышная стратегия, позволяющая ему выиграть третьим ходом при любой игре Вани, но у Пети нет стратегии, которая позволяла бы ему гарантированно выиграть первым или вторым ходом.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241304?t=1h12m5s

Решение

По условию задачи нельзя повторять ход, который этот же игрок делал на предыдущем ходу, т.е. нельзя повторять свой предыдущий ход. Т.е. следует сохранять ходы, которые были совершены игроками, и смотреть на то, какой ход, был на два шага назад. Для этого передадим в функцию параметр tr , который будет обозначать траекторию игры. $f(s,m,tr)$. И в зависимости от значения tr с индексом, указывающим на то, что имеется в виду предпредыдущий ход в игре (или предыдущий ход этого же игрока), заполним список ходов:

```
h = []
if tr[-2]!='1': h.append(f(s+1,m-1,tr+'1'))
if tr[-2]!='2': h.append(f(s+2,m-1,tr+'2'))
if tr[-2]!='3': h.append(f(s*2,m-1,tr+'3'))
```

Для того чтобы вывести на экран результаты работы функции в соответствии с формулировками вопросов следует в качестве третьего параметра функции передать значение p , как '00'. Потому что пустая строка, которую мы использовали в предыдущем задании, в ситуации, когда функции нужно рассмотреть значение на два шага назад, программа будет давать ошибку `string index out of range`, что означает что у нас мы вышли за границы строки, мы берем символ которого не существует. Если же указать '00'. При этом проверка уже будет работать, т.к. в строке уже есть какие-то символы, но при этом они отличаются от тех, что задаются в условии.

```
def f(s,m,tr):
    if s>=29: return m%2==0
    if m==0: return 0
    h = []
    if tr[-2]!='1': h.append(f(s+1,m-1,tr+'1'))
    if tr[-2]!='2': h.append(f(s+2,m-1,tr+'2'))
    if tr[-2]!='3': h.append(f(s*2,m-1,tr+'3'))
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(1,29) if not f(s,1,'00') and f(s,3,'00')])
print(20,[s for s in range(1,29) if not f(s,2,'00') and f(s,4,'00')])
print(21,[s for s in range(1,29) if not f(s,3,'00') and f(s,5,'00')])
```

Результат работы программы:

```
19 [12, 13]
20 [10, 11]
21 [5, 9]
```

Ответ:

```
12
10 11
9
```

Задание № 7(б/н)

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. Если в куче n камней и число n кратно k ($k > 1$), то за один ход разрешается добавить в кучу n/k камней. Например, если в куче 12 камней, то за один ход можно добавить 1 ($12/12$), 2 ($12/6$), 3 ($12/4$), 4 ($12/3$) или 6 ($12/2$) камней. Игра завершается, когда количество камней в куче становится больше 40. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет больше 40 камней. В начале игры в куче было S камней, $S \leq 40$.

19) Укажите количество таких значений S , при которых Петя не может выиграть первым ходом, но при любом первом ходе Пети Ваня может выиграть своим первым ходом.

20) Для игры, описанной в задании 19, найдите наименьшее и наибольшее значения S , при которых Петя не может выиграть первым ходом, но у Пети есть выигрышная стратегия, позволяющая ему выиграть вторым ходом при любой игре Вани.

В ответе запишите найденные значения в порядке возрастания.

21) Для игры, описанной в задании 19, найдите наибольшее значение S , при котором у Вани есть стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, но у Вани нет стратегии, которая позволила бы ему гарантированно выиграть первым ходом.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241304?t=1h26m50s

Решение

Указав условия выхода из функции в генераторе списка запишем ходы игроков. Т.е. мы прибавляем к s все его делители, начиная от единицы кроме самого числа.

```
h = [f(s+s//k, m-1) for k in range(2,s+1) if s%k==0]
```

Этой же записи соответствует запись нижеприведенные закомментированные строки.

Вывод на экран результатов работы функции соответствуют формулировкам вопросов, но следует в качестве диапазона для выбора количеств камней в куче указать нижней границей число 2, т.к. для единиц ходы считаются некорректно, т.к. по условию $k > 1$.

```
def f(s,m):
    if s>40: return m%2==0
    if m==0: return 0
    h = [f(s+s//k, m-1) for k in range(2,s+1) if s%k==0]
    #h = []
    #for k in range(2,s+1):
    #    if s%k==0:
    #        h.append(f(s+s//k,m-1))
    return any(h) if (m-1)%2==0 else all(h)
print(19,[s for s in range(2,41) if f(s,2)])
print(20,[s for s in range(2,41) if not f(s,1) and f(s,3)])
print(21,[s for s in range(2,41) if not f(s,2) and f(s,4)])
```

Результат работы программы:

19 [27, 29, 31, 37]

20 [18, 24, 26]

21 [17, 23, 25]

Ответ:

4

18 26

25

Задание №8 (2365)

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

- а) забрать из кучи один камень;
- б) забрать из кучи два камня;
- в) забрать из кучи четыре камня.

Если камней в куче меньше, чем забирается, то такой ход выполнить нельзя. Игрок, забравший последний камень выигрывает. В начальный момент в куче было S камней, $1 \leq S \leq 15$.

Задание 19. Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите максимальное значение S , когда такая ситуация возможна.

Задание 20.

Найдите два наименьших значения S , при которых у Пети есть выигрышная стратегия, позволяющая ему выиграть вторым или третьим ходом в зависимости от хода Вани, при этом у него нет стратегии, которая позволит ему гарантированно выиграть своим вторым ходом.

Задание 21.

Найдите максимальное значение S , при котором Ваня имеет выигрышную стратегию, при которой он выигрывает при любой игре Пети.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241304?t=1h38m15s

Решение

Для записи ходов следует учесть условие, если камней меньше, чем забирается, то такой ход выполнить нельзя, т.е. для добавления каждого из ходов в список следует проверить, что значение s больше или равно значению хода.

Далее выполним действия для получения ответов на вопросы задачи. Для получения ответа на первый вопрос задачи учтем, что нам необходимо поменять `all` на `any`

Для ответа на третий вопрос задачи в качестве крайнего значения укажем 100, это будет означать «хотя бы за 100 ходов». Здесь важно, указать именно четное число. Это будет означать - победа Вани. Если написать 101, это будет победа Пети

```
def f(s,m):
    if s==0: return m%2==0
    if m==0: return 0
    h = []
    if s>=1: h.append(f(s-1,m-1))
    if s>=2: h.append(f(s-2,m-1))
    if s>=4: h.append(f(s-4,m-1))
    return any(h) if (m-1)%2==0 else all(h)

#all->any
print(19,[s for s in range(1,16) if f(s,2)])
print(20,[s for s in range(1,16) if not f(s,3) and f(s,5)])
print(21,[s for s in range(1,16) if f(s,100)]) #хотя бы за сто ходов
```

Результат работы программы:

19 [2, 3, 4, 5, 6, 8]

20 [8, 10]

21 [3, 6, 9, 12, 15]

Важно! Если в каких то задачах вычисление происходит медленно, следует подключить кэш.

Ответ:

8

8 9

15

Задание №9(1158)

Два игрока, Паша и Вероника, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может из каждой кучи убрать по три камня или убрать целиком одну кучу, а другую разделить на две равные (если это позволяет количество камней).

Игра завершается после того хода, когда хотя бы одна куча становится пустой или когда невозможно сделать очередной ход по правилам. Победителем считается игрок, сделавший последний ход. В начальный момент времени в одной куче лежит N камней, а в другой – K камней.

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

Задание 19. Известно, что после неудачного первого хода Паши Вероника выиграла первым своим ходом. При каком наибольшем значении K это возможно, если $N = 32$?

Задание 20. Для игры, описанной в задании 19, при $N = 32$ найдите такие значения K , при которых у Паши есть выигрышная стратегия, причём Паша не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Вероника.

Из всех найденных значений запишите в ответе минимальное и максимальное в порядке возрастания.

Задание 21. Для игры, описанной в задании 19, при $N = 20$ найдите наибольшее значение K , при котором у Вероники есть выигрышная стратегия, позволяющая ей выиграть своим первым или вторым ходом.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241304?t=1h48m25s

Решение

В этой задаче интересными являются как условие выхода из игры так и описание условий ходов.

Первое условие, когда обе кучи становятся пустыми реализовать можно, записав:

```
if a==0 or b==0
```

Для реализации второго условия предварительно мы должны создать список ходов, указав их варианты, в зависимости от ситуаций игры.

```
if a>=3 and b>=3: h.append(f(a-3,b-3,m-1))
if a%2==0: h.append(f(a//2, a//2, m-1))
if b%2==0: h.append(f(b//2,b//2,m-1))
```

Ходы, в которых одна куча убирается, а вторая делится пополам можно записать, как $a//2$, записанное дважды, т.к. в качестве второй кучи (b) становится половина первой кучи. Заметим, что игра заканчивается победой, когда у игрока нет вариантов ходов, следовательно, в списке ходов не окажется ни одного элемента, или другими словами, его длина будет равна 0.

```
if len(h)==0: return m%2==0
```

Завершаем написание функции обычным образом. И записываем строки для получения ответов на вопросы задачи. Для получения ответа на первый вопрос задачи учтем, что нам необходимо поменять `all` на `any`

```
def f(a,b,m):
    if a==0 or b==0: return m%2==0
    h = []
    if a>=3 and b>=3: h.append(f(a-3,b-3,m-1))
    if a%2==0: h.append(f(a//2, a//2, m-1))
    if b%2==0: h.append(f(b//2,b//2,m-1))
    # если ходов нет, то game over
    if len(h)==0: return m%2==0
    if m==0: return 0
    return any(h) if (m-1)%2==0 else all(h)

#all->any
print(19, [k for k in range(1,100) if f(32,k,2)])
print(20, [k for k in range(1,100) if not f(32,k,1) and f(32,k,3)])
print(21, [k for k in range(1,100) if not f(20,k,2) and f(20,k,4)])
```

Результат работы программы:

19 [4, 5, 6, 8]

20 [8, 9, 10, 12, 14]

21 [1, 5, 6, 13, 15, 17]

Ответ:

8

8 14

17

Задание №10 (584)

Два игрока, Петя и Ваня, играют в игру. В начале игры перед игроками лежит игровая кость (кубик с точками на гранях, количество точек от 1 до 6). Начальной позицией считается грань, находящаяся сверху. Игроки совершают ходы по очереди, первый ход делает Петя.

За один ход игрок может положить кость, на верхней грани которой будет на одну точку больше или на одну точку меньше. Для граней с 1 и 6 точками возможен только 1 ход – 2 точки и 5 точек, соответственно. Выигрывает игрок, после хода которого суммарное количество точек на всех выложенных костях будет не меньше N .

Найдите все значения N , при которых Ваня выигрывает своим первым ходом при любой игре Пети, если известно, что игра начинается с грани с 5 точками. В качестве ответа укажите два значения – минимальное и максимальное значения N .

Задание 20. Для условия игры из задания 19, ответьте на вопрос.

Игра начинается с грани с 4 точками. Найдите минимальное и максимальное значения N , при которых у Пети есть выигрышная стратегия в два хода.

Задание 21. Для условия игры из задания 19, ответьте на вопрос.

Игра начинается с грани с 1 точкой. Найдите минимальное и максимальное значения N , при которых у Вани есть выигрышная стратегия в два хода

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241304?t=1h59m40s

Решение

В этой задаче неизвестным является условие окончания игры, но известно, когда она начинается, т.е. в ходе решения будет необходимо подобрать значение окончания игры. Так же будет необходимо контролировать условия тех или иных ходов.

В качестве параметров будем передавать функции список костей ($kosti$), номер хода (m) и параметр end , условие окончания игры. $kosti$ – это список с числами от 1 до 6, обозначающими количества точек на выпавших гранях кубиков. Условием первой победы будет значение суммы на костях больше значения условия окончания игры.

Запишем варианты списков ходов в зависимости от того, каким был предыдущий ход соперника. При этом в список $kosti$ мы будем добавлять значение хода в виде элемента списка. Предыдущее значение мы обозначим переменной $pred$.

После того, как будет написана функция, определяющая ход игры, записываем строки для получения ответов на вопросы задачи. Для ответа на первый вопрос задачи. Следует указать диапазон от 6, т.к. минимум 5 уже есть. Далее проверяем f от 5, мы начинаем с одной пятерки, т.е. рассматриваем грань с пятью точками, побед должна быть за 2 хода и заканчиваться за n камней. По аналогии напишем строки для получения ответов на 2 и третий вопрос задачи, указывая в качестве начала диапазона значение на единицу большее, чем количество точек, которые игрок должен выставить в качестве хода.

```
def f(kosti,m,end):
    pred = kosti[-1]
    if sum(kosti)>=end: return m%2==0
    if m==0: return 0
    if pred==1: h = [f(kosti+[2],m-1,end)]
    if pred==6: h = [f(kosti+[5],m-1,end)]
    if 1<pred<6: h = [f(kosti+[pred+1],m-1,end), f(kosti+[pred-1],m-1,end)]
    return any(h) if (m-1)%2==0 else all(h)
```



```
print(19,[n for n in range(6,100) if f([5],2,n)])  
print(20,[n for n in range(5,100) if not f([4],1,n) and f([4],3,n)])  
print(21,[n for n in range(2,100) if not f([1],2,n) and f([1],4,n)])
```

Результат работы программы:

19 [12, 13, 14]

20 [12, 16, 17, 18]

21 [7, 8, 9, 11]

Ответ :

12 14

12 18

7 11

Telegram: @fast_ege