

Strim_19_20_21_pr1

Задача № 1 (3077)

(А. Кабанов) 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 два камня или увеличить количество камней в куче в три раза. Например, имея кучу из 10 камней, за один ход можно получить кучу из 12 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 50. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 50 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 49$.

Ответьте на следующие вопросы:

Вопрос 1. Найдите минимальное значение S , при котором Ваня выигрывает своим первым ходом при любой игре Пети.

Вопрос 2. Сколько существует значений 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.

Вопрос 3. Найдите два значения S , при которых 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. Найденные значения запишите в ответе в порядке возрастания.

Решение

Как работает функция определения стратегии: https://vk.com/video-205546952_456241300?t=0h3m30s

Начнём с написания функции, которая будет определять, существует ли стратегия победы из определенного значения s за определенное количество ходов. Т.е., будем передавать в функцию значения начального количества камней и количество ходов, за которые мы хотим победить. И, затем, функция будет выяснять, можно ли выиграть за это количество ходов.

Это функция, которая принимает два параметра. s — это начальное количество камней, m — желаемое нами количество ходов.

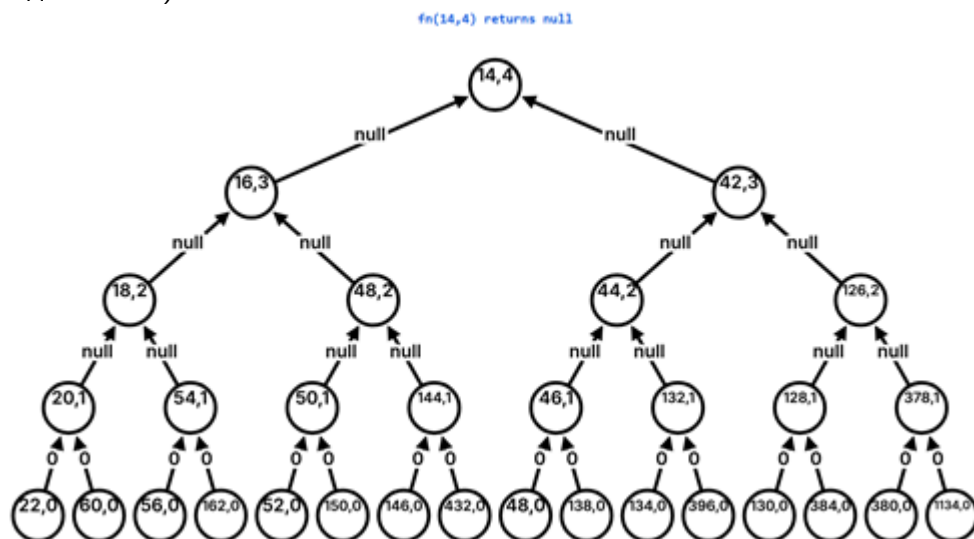
Если m равно 0, останавливается и возвращает 0 (m будет уменьшаться с каждым вызовом). h — это список, в котором записаны результаты вызовов функции от s с учётом заданных по условию задачи ходов:

- $s+2$ — ход добавить 2 камня
- $s*3$ — ход увеличить кучу в 3 раза,

```
def f(s, m):
```

```
if m==0: return 0
h = [f(s+2,m-1), f(s*3,m-1)]
```

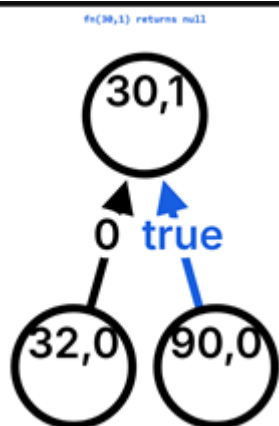
С помощью такой функции можно построить полное дерево игры, в зависимости от начального количества камней в куче и заданного общего количества ходов до победы. Получим визуализацию работы этой функции с помощью приложения <https://recursion.vercel.app/>. Зададим начальное количество камней в куче равным 14, а количество ходов 4 (победа вторым ходом Вани)



Обратим внимание, что полученное дерево, не отражает условия задачи о том, что игра заканчивается, что как только количество камней достигает 50, дополним нашу функцию этим условием:

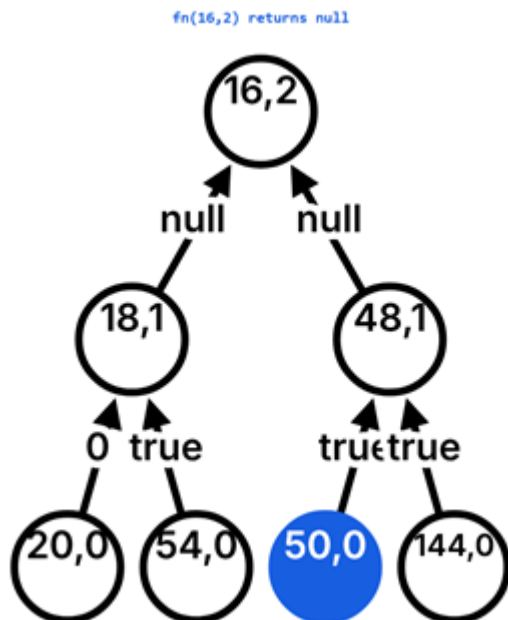
```
if s>=50: return m==0
```

Это условие, которое позволит выделить такие траектории, в которых игра заканчивается ровно за m ходов. Визуализируем вариант игры со стартовым значением $s=30$ и количеством ходов $m=1$ (победа Пети своим первым ходом):

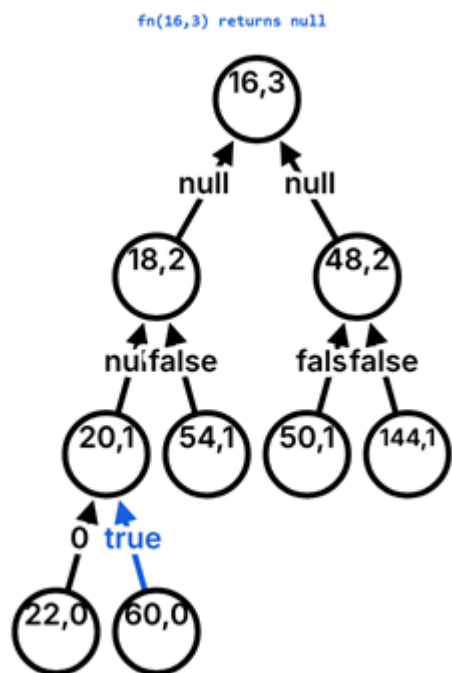


Это происходит в тот момент, когда счетчик m становится равным 0. Вариант, в котором игра закончилась ровно за один ход, отмечен как `true`.

Пример варианта игры со значениями $s=16$ и $m=2$:



Победа происходит ровно на втором ходу – функция вернула нам значение True. Если взять значение $m=3$, т.е. победа должна быть получена за три хода, то такие варианты уже не будут True, потому что произошли раньше, чем нужно.



Победный вариант в данном случае будет единственным, а предыдущие ходы вернут значение False. Т.е. с помощью этого кода

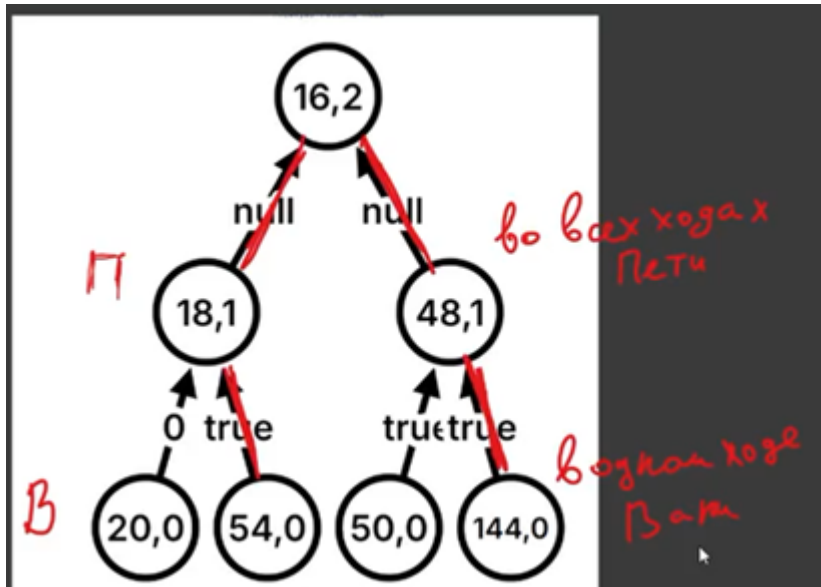
```
def f(s,m):
    if s>=50: return m==0
    if m==0: return 0
    h = [f(s+2,m-1),f(s*3,m-1)]
```

Мы можем получить все варианты ходов при которых происходит победа за указанное количество ходов.

Теперь на основании этих траекторий нам следует оценить существует ли выигрышная стратегия для того или иного игрока?

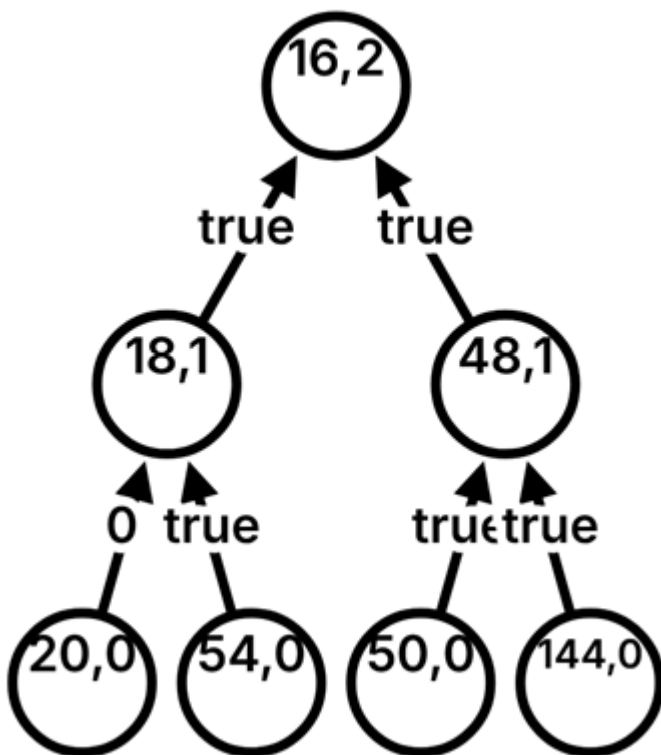
Рассмотрим пример со стартовым числом камней в куче равным 16 и победой Вани вторым ходом ($m=2$), т.к. это число из которого Ваня гарантированно выиграет своим первым ходом при любом

ходе Пети.



Видим, что в этом случае значение m соответствующее ходу Пети равно единице, а у Вани m равно нулю. Т.е. у Пети значение четное, у Вани оно нечетное. Обратим внимание, что, когда смотрим такую стратегию, мы проверяем её во всех ходах Пети, и только в одном ходе Вани.

`fn(16,2) returns true`



Т.е. стратегия существует, если она работает при любом ходе Пети и хотя бы в одном ходе Вани. Это можно записать как:

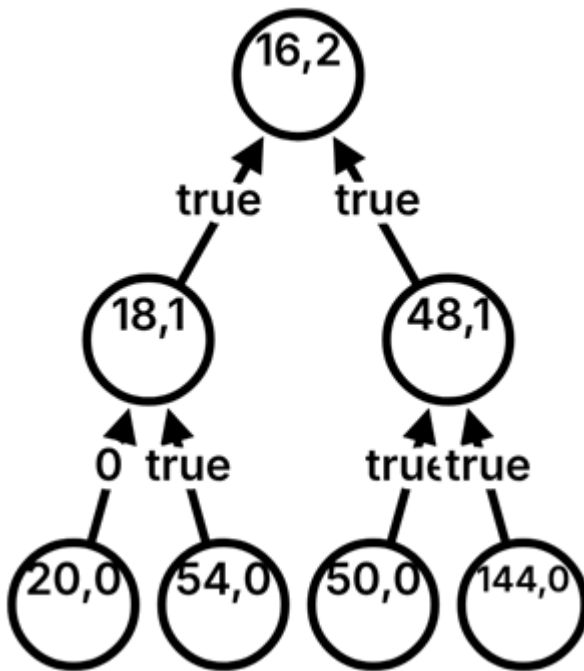
```
return any(h) if (m-1)%2==0 else all(h)
```

`any(h)`- любое значение из списка ходов

`m-1`- это значение m в следующем ходе, т.к. когда делается ход, счетчик уменьшается на единицу. Следовательно, если $(m-1)\%2$ имеет четное значение – это ход Вани и мы возвращаем любой его ход – он будет выигрышным, а если нечетное значение, значит, это ход Пети. И так как нам требуется получить выигрыш при всех вариантах всех ходов Пети, мы должны записать `all(h)`

Т.е. если в одном из ходов Вани при всех ходах Пети есть победа, значит, значение подходит. Проверим число 15.

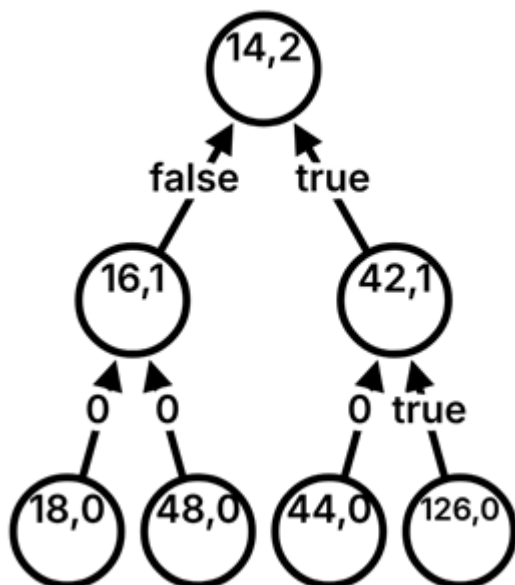
`fn(16,2) returns true`



Здесь в одном из ходов Вани есть победа. И во всех ходах Пети тоже получается такая ситуация. Значит у Вани существует выигрышная стратегия, позволяющая ему выиграть своим вторым ходом.

Проверим число 14.

`fn(14,2) returns false`



Очевидно, что у Вани есть вариант выигрышного хода, но не при всех вариантах ходов Пети, т.е. мы уже не можем говорить о наличии у него выигрышной стратегии.

Иначе мы можем записать это так:

```
def f(s,m):  
    if s>=50: return m==0  
    if m==0: return 0
```

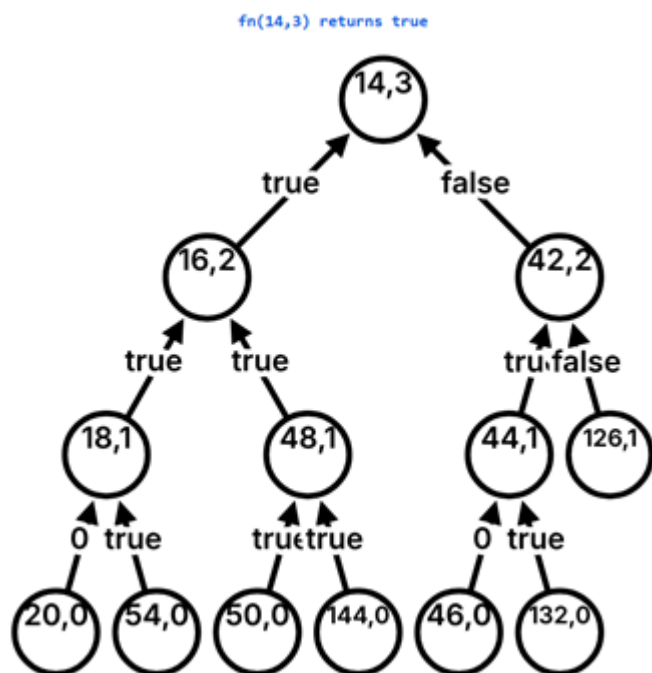
```

# Ходы следующего игрока
h = [f(s+2,m-1), f(s*3,m-1)]
# Если ход совершает Ваня, то выигрыш должен быть в одном из его ходов
if (m-1)%2==0: return any(h)
# Если ход совершает Петя, то выигрыш Вани должен быть при всех его ходах
else: return all(h)

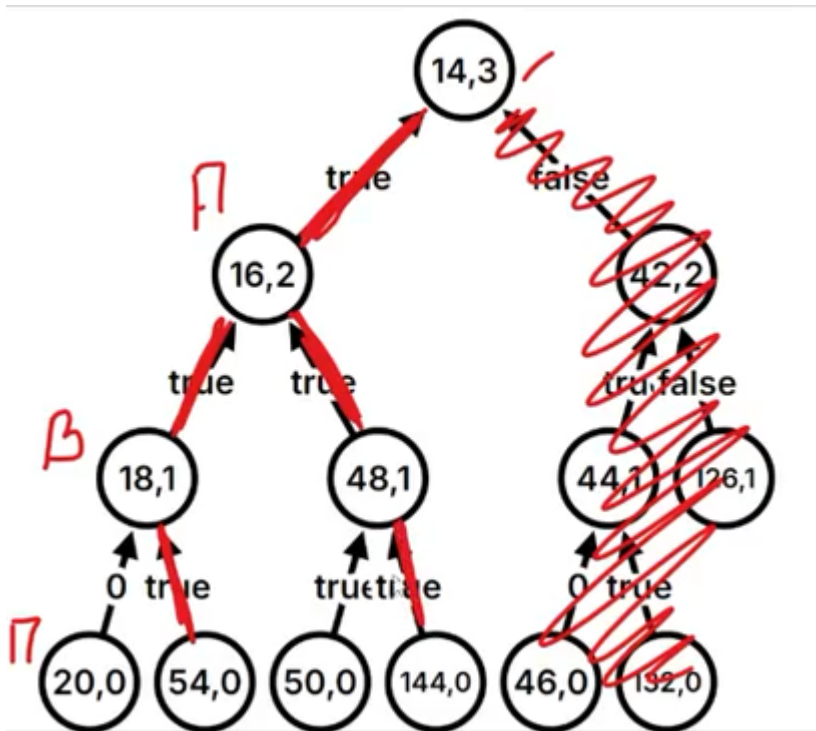
```

Т.е., смотрим значение функций в следующих ходах, и в зависимости от того, что за игрок, у нас должна быть стратегия в одном из этих ходов или во всех этих ходах. Так, Петя «нечётный», и нам следует просмотреть все его ходы, а Ваня «чётный», и нам подходит даже одна выигрышная стратегия.

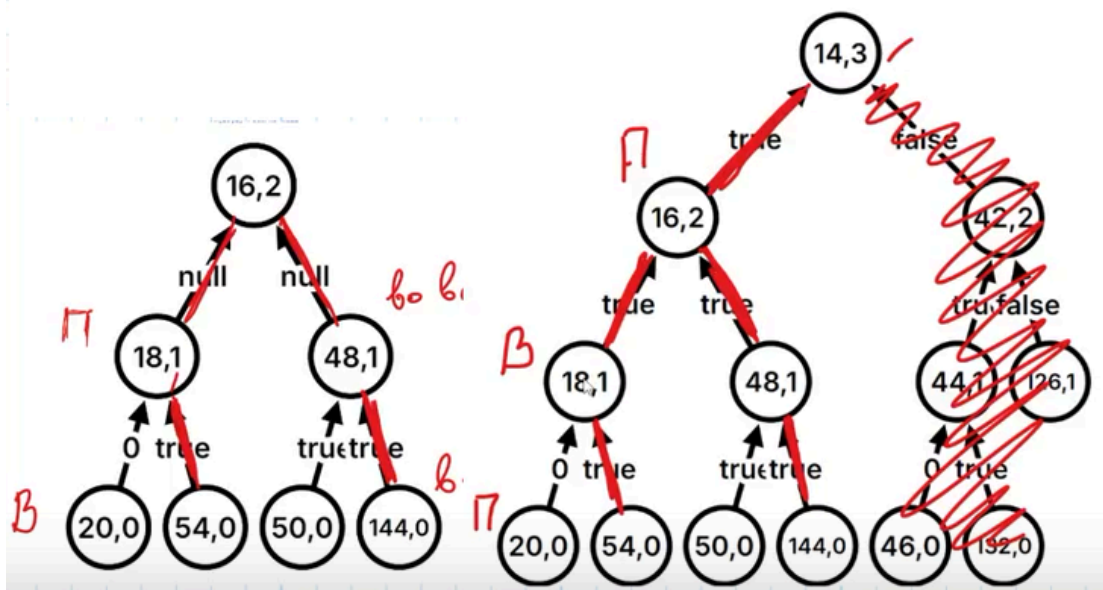
Рассмотрим вариант задачи с победой за три хода. Можно предположить, что подходящим будет число 14. Так будет выглядеть древо игры с этими входными параметрами:



Эту стратегию мы можем проследить по стрелкам с надписью True. Т.е. Пети есть такой ход, в 16, при котором как бы после него Ваня не ходил, Петя выигрывает. Т.е. это стратегия, в которой Петя гарантированно побеждает своим вторым ходом.

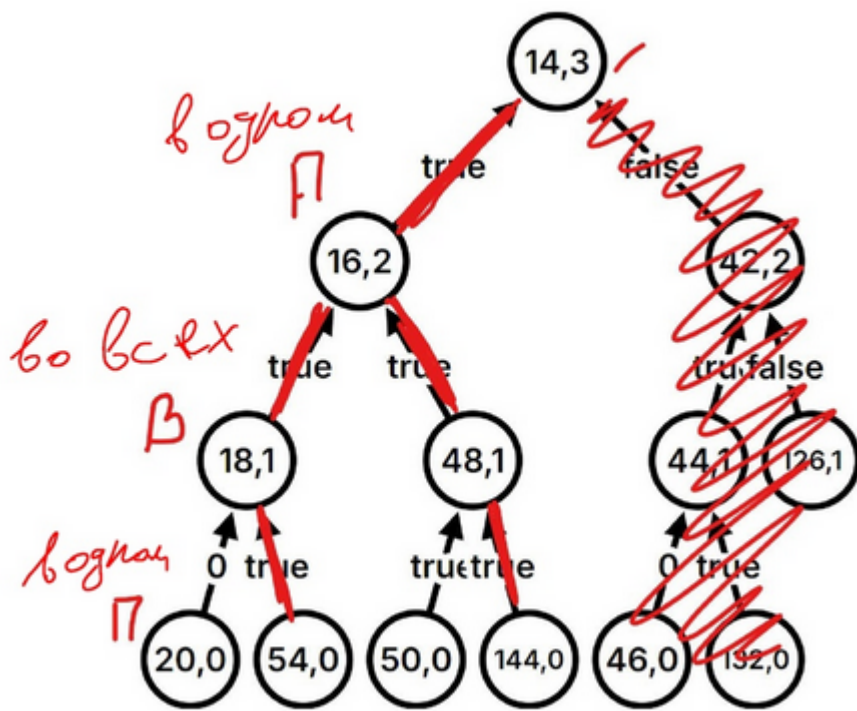


Сравним изображение двух деревьев рассмотренных примеров:



Заметим, что Петя стал «чётный», а Ваня стал «нечётный». Т.е., они поменялись местами. Это произошло потому, что мы передали в нашу функцию нечетный параметр. Во втором примере работа функции началась с нечетного числа 3.

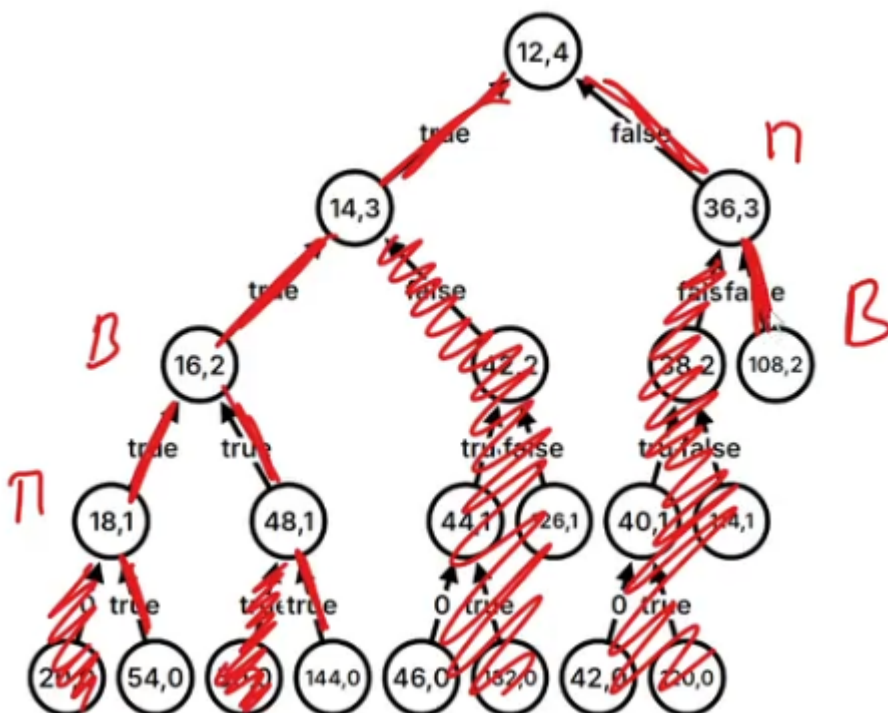
Т.е., там где значение m будет чётное, победы достаточно в одном из ходов. Там, где m нечётное, там победа должна быть во всех ходах:



И это самый интересный момент вот этой программы. Он заключается в том, что в зависимости от того, какое m у вас задано в самом начале, если оно чётное, значит вы ищите стратегию для Вани, если нечётное, значит вы ищите стратегию для Пети. Задано 1 или 3, значит вы для Пети ищите. Задано 2 или 4, значит для Вани.

Назовём игрока, для которого нужно получить победу целевой игрок. Т.е. это игрок, для которого мы ищем стратегию. Если параметр m задается нечетным числом, 1, 3, или 5, целевым игроком будет Петя. Если число четное 2 или 4 – это игрок Ваня. Вот. Если ход совершает соперник, то стратегия должна быть во всех его ходах.

Следует обратить внимание на вопрос когда требуется определить победу игрока первым или вторым ходом. Рассмотрим пример, когда победа Вани первым или вторым ходом при начальном количестве камне в куче равном 12.



Действительно из 12 можно выиграть первым или вторым ходом, но программа демонстрирует выигрышную стратегию только для 4х ходов (видим True). Потому что как бы ни ходил Петя первым ходом, Ваня делает выигрышный ход в 16. И дальше, как бы ни ходил опять же Петя, Ваня выигрывает. Это стратегия Вани. Обратим внимание, что есть еще один вариант игры, при котором Ваня тоже выигрывает, но раньше. Очевидно, что написанная нами, функция проверяет победу только ровно за 4 хода, а нам необходимо, чтобы функция учитывала победу не только за 4, но и за меньшее количество ходов. Заметим, что в древе на победном уровне, т.е. на четвертом ходу $m = 0$, на являющемся также победном 2 уровне игры, значение m равно 2. Видим, что оба значения m являются четными. Т.е. мы можем выиграть менее, чем за 4 хода, но, в любом случае это должен быть Ваня. Т.е. допускается победа раньше, но тем же самым игроком, у которого значение m той же четности.

Т.е. в условии вместо того, чтобы $m == 0$, можем написать $m == 0$ или $m == 2$.

```
if s >= 50: return m == 0 or m == 2
```

Более же универсальный подход будет заключаться в том, что мы будем проверять четность значения m . Т.е. если значение m четное, значит побеждает тот игрок, которому ищем стратегию

```
if s >= 50: return m % 2 == 0
```

Теперь имея этот универсальный ход, рассмотрим решение нашей конкретной задачи.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241300?t=0h41m25s

Уточним нумерацию ходов:

P1 – 1

V1 – 2

P2 – 3

V2 – 4

1, 2, 3, 4 – это значение параметра m , определяющего глубину уровней ходов.

Для получения ответа на первый вопрос задачи запишем следующую строку:

```
print(19, [s for s in range(1, 50) if f(s, 2)])
```

Здесь мы выведем все значение s , из диапазона от 1 до 49, если из них существует стратегия победы за 2 хода.

Запишем строку для получения ответа на второй вопрос задачи, учитывая оба поставленных условия:

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

```
print(20, [s for s in range(1, 50) if not f(s, 1) and f(s, 3)])
```

Т.е. здесь мы выведем все значение s , из диапазона от 1 до 49, в которых не существует стратегии выигрыша за один ход и при этом существует победа за 3 хода.

Для ответа на третий вопрос задачи учтем условие:

- 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.

```
print(21, [s for s in range(1, 50) if not f(s, 2) and f(s, 4)])
```

В данной строке мы видим, что Ваня НЕ может гарантировано выиграть своим первым ходом ($\text{not } f(s,2)$), но может гарантировано выиграть своим вторым ходом $\text{and } f(s,4)$.

```
def f(s,m):
    if s>=50: return m%2==0
    if m==0: return 0
    h = [f(s+2,m-1), f(s*3,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(1,50) if f(s,2)])
print(20,[s for s in range(1,50) if not f(s,1) and f(s,3)])
print(21,[s for s in range(1,50) if not f(s,2) and f(s,4)])
```

Результат работы программы:

```
19 [15, 16]
20 [5, 13, 14]
21 [11, 12]
```

Ответ: 15
3
11 12

Задача №2 (7625)

(Демо-2025) 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.

Ответьте на следующие вопросы:

Вопрос 1. Укажите минимальное значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.

Вопрос 2. Найдите два наименьших значения S , когда Петя имеет выигрышную стратегию, 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 3. 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_456241300?t=0h49m35s

Решение

Напишем функцию с учетом заданного условия победы, s должно быть не больше 19. В список ходов занесем результаты работы функции в зависимости от возможного варианта хода. За один ход игрок может: убрать из кучи два камня ($s-2$) или убрать из кучи пять камней ($s-5$) или уменьшить количество камней в куче в три раза (количество камней, полученное при делении, округляется до меньшего, т.е. нам следует использовать целочисленное деление, $s//3$).

Выведем на экран значения необходимые для ответа на вопросы задачи.

19. Т.к. в задаче нет верхнего предела рассмотрим значение s из диапазона от 20 до 1000.

Для ответа на 20 вопрос нам нужно, чтобы Петя не мог выиграть первым ходом, но Петя мог выиграть своим вторым ходом т.е., в общем счете ходов номер победного хода будет 3.

Задача 21. Ваня не может выиграть первым ходом, но Ваня может выиграть самым первым или вторым.

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

print(19,[s for s in range(20,1000) if f(s,2)])
print(20,[s for s in range(20,1000) if not f(s,1) and f(s,3)])
print(21,[s for s in range(20,1000) if not f(s,2) and f(s,4)])
```

Результат работы программы:

19 [60, 61]

20 [62, 63, 65, 66, 180, 181, 182, 183, 184, 185]

21 [64, 67, 68, 186, 187]

Ответ:

60

62, 63

64

Задача № 3 (3082)

(А. Кабанов) 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) добавить в кучу два камня;

б) увеличить количество камней в куче в три раза.

Игра завершается в тот момент, когда количество камней в куче становится не менее 45. Если при этом в куче оказалось не более 112 камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его противник. В начальный момент в куче было S камней, $1 \leq S \leq 44$.

Ответьте на следующие вопросы:

Вопрос 1. Найдите минимальное значение S , при котором Ваня выигрывает своим первым ходом при любой игре Пети.

Вопрос 2. Сколько существует значений 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.

Вопрос 3. Найдите минимальное и максимальное значения S , при которых 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.

Найденные значения запишите в ответе в порядке возрастания.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241300?t=0h56m35s

Решение

Отметим, что если S выпадает в диапазоне от 45 до 112 – это победа. И она должна попасть в m целевого игрока, в четное m . Т.е. если $s > 112$ и это поражение и ход противника, поэтому мы это запишем как $m\%2 \neq 0$. В списке ходов укажем заданные по условию варианты $s+2$ и $s*3$.

С учетом заданного порядка ходов выведем на экран значения для ответов на вопросы 19, 20, 21

```
def f(s,m):
    if 45<=s<=112: return m%2==0
    if s>112: return m%2!=0
    if m==0: return 0
    h = [f(s+2,m-1),f(s*3,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(1,45) if f(s,2)])
print(20,[s for s in range(1,45) if not f(s,1) and f(s,3)])
print(21,[s for s in range(1,45) if not f(s,2) and f(s,4)])
```

Результат работы программы:

19 [41, 42]

20 [14, 39, 40]

21 [12, 13, 38]

Ответ:

41

3

12 38

Задача № 4 (2414)

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 55. Победителем считается игрок, сделавший последний ход, т. е. первым получивший позицию, в которой в кучах будет 55 или больше камней. В начальный момент в первой куче было 9 камней, во второй куче – S камней, $1 \leq S \leq 45$. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Ответьте на следующие вопросы:

Вопрос 1. Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Назовите минимальное значение S , при котором это возможно.

Вопрос 2. Укажите минимальное значение S , при котором 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.

Вопрос 3. Найдите два значения S , при которых у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и при этом у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Найденные значения запишите в ответе в порядке возрастания

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241300?t=1h4m40s

Решение

Обратим внимание, что в первом вопросе указано, что Ваня выиграл своим первым ходом не при любом ходе Пети, т.е. не во всех ходах Пети, а в каком-то неудачном.

Для решения этой задачи мы также напишем функцию, в которой передадим

информацию не об одной куче, а о двух. Для этого внутри функции создаем две переменные.

Назовем эти переменные a и b . a и b — это первая и вторая куча камней, соответственно. Далее, запишем все необходимые строки учитывая эти два параметра. В списке ходов укажем 4 варианта хода, которые у нас есть для обеих куч. И далее следует, как и в предыдущих задачах вывести на экран значения в зависимости от порядка выигрышных ходов.

```
def f(a,b,m):
    if a+b>=55: return m%2==0
    if m==0: return 0
    h = [f(a+2,b,m-1), f(a,b+2,m-1), f(a*2,b,m-1), f(a,b*2,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

#all->any
print(19,[s for s in range(1,46) if f(9,s,2)])
```

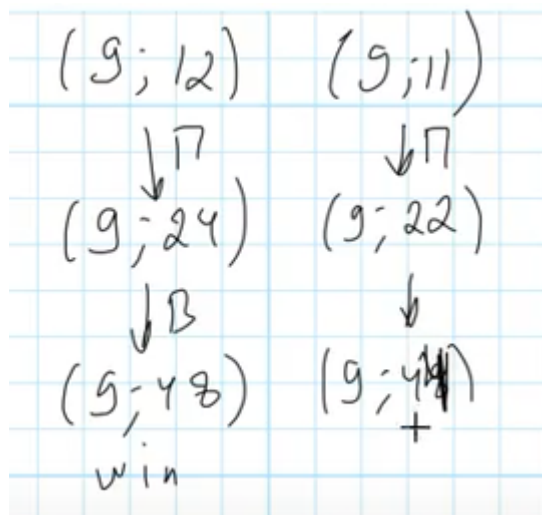
Результат работы программы:

19 [22]

Но если мы для ответа на первый (19) вопрос задачи запустим программу, то получим одно и неверное значение 22. Это произошло потому, что мы нашли значение, да, s , из которого Ваня выигрывает при любой игре Пети. Во всех четырех ходах, как бы Петя ни сходил, Ваня выиграет. Но по условию задачи требуется чтобы Ваня выиграл в каком-то одном неудачном ходе. Как отследить с помощью программы этот неудачный ход? За это отвечает `all`, который говорит, что вот во всех ходах противника должны выигрывать, во всех ходах противника должна существовать выигрышная

стратегия. А если поменять all на any, то тем самым мы найдём победу не во всех ходах, а в каком-то одном, т.е. есть какой-то один ход, из которого потом Ваня выигрывает. Если мы сделаем такую замену, то значений становится значительно больше. От 12 до 43.

Аналитически проверим возможность выигрыша Вани при каком-то ошибочном ходе противника:



Видим, что при условии, если в одной из куч будет 12 камней и Петя сделает неудачный для него ход (9;14), Ваня своим следующим ходом получит победу, но уже в случае, если в куче будет 11 камней, Ваня не сможет получить выигрыш, даже при неудачном ходе Пети, т.к. ему просто не хватит камней.

Таким образом, для ответа на вопрос 19 данной задачи, т.е. при наличии условия выигрыша после неудачного хода противника в нашей программе следует заменить all на any.

Вопросы 20 и 21 будут решаться обычным образом.

```

def f(a,b,m):
    if a+b>=55: return m%2==0
    if m==0: return 0
    h = [f(a+2,b,m-1), f(a,b+2,m-1), f(a*2,b,m-1), f(a,b*2,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

#all->any
print(19, [s for s in range(1,46) if f(9,s,2)])
print(20, [s for s in range(1,46) if not f(9,s,1) and f(9,s,3)])
print(21, [s for s in range(1,46) if not f(9,s,2) and f(9,s,4)])
  
```

Результат работы программы:

19 [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43]

20 [11, 17, 18, 20, 21]

21 [9, 19]

Ответ:

12

11

9 19

Задача №5 (3791)

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в любую кучу один камень или добавить в любую кучу столько камней, сколько их в данный момент в другой куче. Игра завершается в тот момент, когда общее количество камней в двух кучах становится не менее 81. Победителем считается игрок, сделавший последний ход. В начальный момент в первой куче было 7 камней, а во второй – S камней, $1 \leq S \leq 73$. Ответьте на следующие вопросы:

Вопрос 1. Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Назовите минимальное значение S , при котором это возможно.

Вопрос 2. Найдите минимальное и максимальное значение S , при котором 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 3. Найдите значение S , при котором одновременно выполняются два условия:

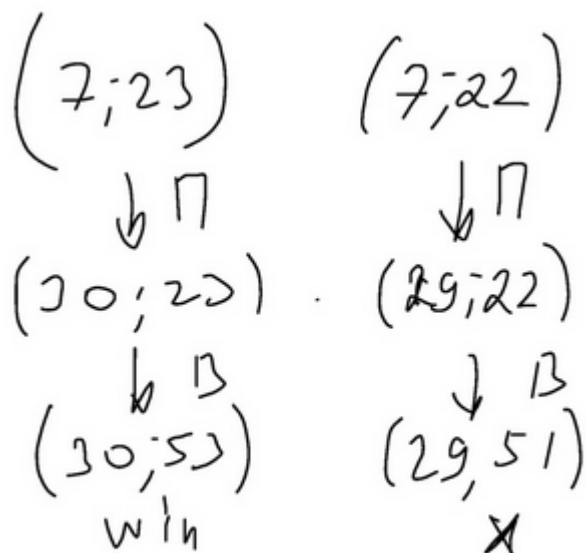
- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241300?t=1h16m40s

Решение

Ход решения задачи полностью аналогичен решению предыдущей задачи.

Ручной анализ ситуации выигрыша Вани при неудачном ходе Пети:



```
def f(a, b, m):
    if a+b>=81: return m%2==0
    if m==0: return 0
    h = [f(a+1,b,m-1), f(a,b+1,m-1), f(a+b,b,m-1), f(a,b+a,m-1)]
    return any(h) if (m-1)%2==0 else all(h)
```



```
#all->any
print(19,[s for s in range(1,74) if f(7,s,2)])
print(20,[s for s in range(1,74) if not f(7,s,1) and f(7,s,3)])
print(21,[s for s in range(1,74) if not f(7,s,2) and f(7,s,4)])
```

Результат работы программы :

```
19 [23...72]
20 [22, 36]
21 [35]
```

Ответ:

```
23
22 36
35
```

Задание №6 (6647)

(Е. Джобс) 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) два камня, или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 5 камней; такую позицию в игре будем обозначать $(10, 5)$. Тогда за один ход можно получить любую из четырёх позиций: $(12, 5)$, $(20, 5)$, $(10, 7)$, $(10, 10)$. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда произведение количеств камней в кучах становится не менее 123. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, при которой произведение числа камней в кучах будет 123 или более.

В начальный момент в первой куче было 3 камня, во второй куче - S камней; $1 \leq S \leq 40$.

Ответьте на следующие вопросы:

Вопрос 1. Найдите наибольшее значение S , при котором Ваня выигрывает своим первым ходом после неудачного хода Пети.

Вопрос 2. Найдите два наибольших 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 3. Найдите наибольшее 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;
- Петя может выбрать, каким ходом выиграет Ваня

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241300?t=1h23m20s

Решение

Первые два вопроса задачи могут быть решены абсолютно аналогично предыдущим примерам. Для ответа на вопрос 19, нам следует поменять all на

any.

```
def f(a,b,m):
    if a*b>=123: return m%2==0
    if m==0: return 0
    h = [f(a+2,b,m-1),f(a,b+2,m-1),f(a*2,b,m-1),f(a,b*2,m-1)]
    return any(h) if (m-1)%2==0 else any(h)

print(19,[s for s in range(1,41) if f(3,s,2)])
```

Результат работы программы:

19 [11... 38]

Победа Вани при неудачном ходе Пети:

(3;38)
↓ П
(3;40)
↓ В
(5;40)
win

```
def f(a,b,m):
    if a*b>=123: return m%2==0
    if m==0: return 0
    h = [f(a+2,b,m-1),f(a,b+2,m-1),f(a*2,b,m-1),f(a,b*2,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print(20,[s for s in range(1,41) if not f(3,s,1) and f(3,s,3)])
print(21,[s for s in range(1,41) if not f(3,s,2) and f(3,s,4)])
```

Результат работы программы:

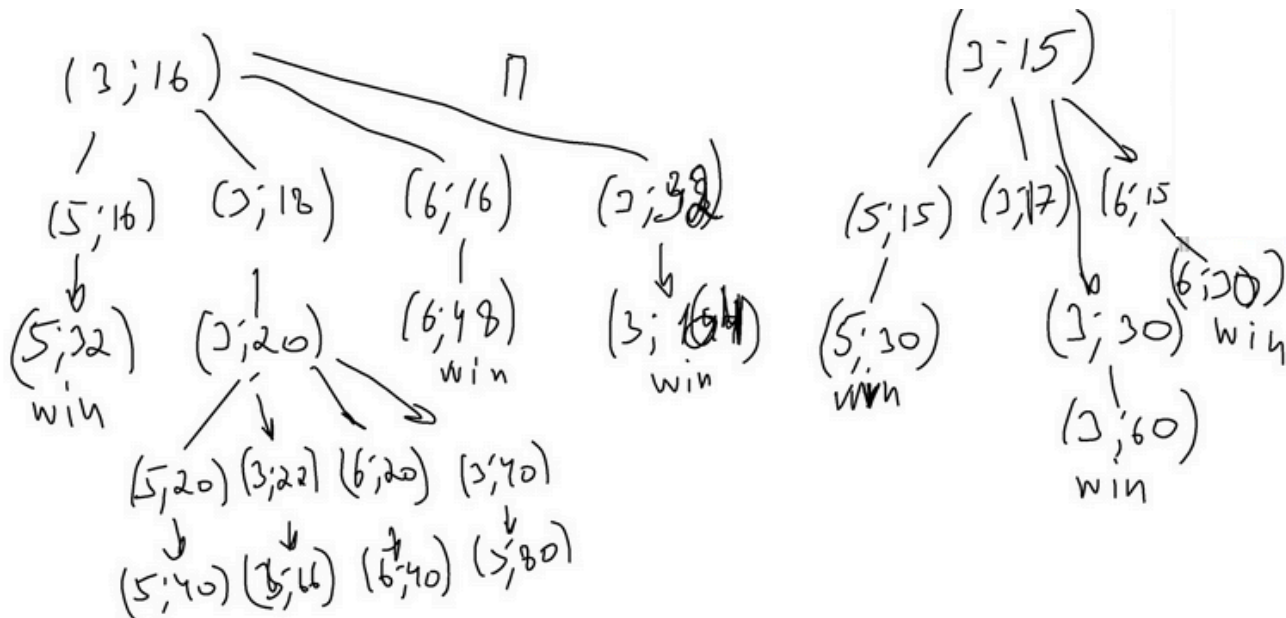
20 [9, 10, 11, 12, 17, 18]

21 [15, 16]

Разберём решение третьего вопроса вручную.

Нам нужно, чтобы Петя мог выбирать, каким ходом выиграет Ваня. При игре из (3;16) Петя может сходить в (3;18) и тогда Ваня сможет выиграть только своим вторым ходом, если же Ваня сходит в любую из позиций (5;16), (6; 16) и (3; 32), то Ваня выигрывает вторым ходом.

Следовательно, для (3;16) выполняется условие, что Петя может выбрать, каким ходом будет выигрывать Ваня.



Получаем ответ 16.

Ответ:

- 1) 38
- 2) 17 18
- 3) 16

Задание № 7(3085)

(А. Кабанов) 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 из одной из куч один камень или уменьшить количество камней в куче в два раза (если количество камней в куче нечётно, остаётся на 1 камень больше, чем убирается). Например, пусть в одной куче 6, а в другой 9 камней; такую позицию мы будем обозначать $(6, 9)$. За один ход из позиции $(6, 9)$ можно получить любую из четырёх позиций: $(5, 9)$, $(3, 9)$, $(6, 8)$, $(6, 5)$. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не более 20. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 20 или меньше камней. В начальный момент в первой куче было 10 камней, во второй куче – S камней, $S > 10$.

Ответьте на следующие вопросы:

Вопрос 1. Найдите значение S , при котором Ваня выигрывает своим первым ходом при любой игре Пети.

Вопрос 2. Найдите минимальное и максимальное значение S , при котором 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 3. 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_456241300?t=1h39m40s

Решение

Обратим внимание, что здесь важный момент, когда делим на два, если количество нечетное, то результат округляется в большую сторону. Для этого будет удобно использовать функцию `ceil`. Это функция округления только в большую сторону. Эта функция берется из модуля `math`.

$$9:2 = 5$$

$$7/10$$

$9//2 = 4$ - округление в меньшую
 $\text{ceil}(9/2)$ - округление в большую.

В остальном в решении задачи будут использованы приемы полностью аналогичные решению вышеразобранных задач.

```
from math import ceil

def f(a,b,m):
    if a+b<=20: return m%2==0
    if m==0: return 0
    h = [f(a-1,b,m-1),f(a,b-1,m-1),f(ceil(a/2),b,m-1),f(a,ceil(b/2),m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(11,1000) if f(10,s,2)])
print(20,[s for s in range(11,1000) if not f(10,s,1) and f(10,s,3)])
print(21,[s for s in range(11,1000) if not f(10,s,2) and f(10,s,4)])
```

Результат работы программы:

```
19 [21]
20 [22, 23, 31, 41, 42]
21 [24]
```

Ответ:

- 1) 21
- 2) 22 42
- 3) 24

Задание №8 (6771)

(А. Рогов) Два игрока, Петя и Ваня, играют в следующую игру. На координатной плоскости стоит фишка. Игроки ходят по очереди. Ход состоит в том, что игрок перемещает фишку из точки с координатами (x, y) в одну из трех точек: или в точку с координатами $(2x, y)$, или в точку с координатами $(x, y+3)$, или в точку с координатами $(x, y+4)$. Выигрывает игрок, после хода

которого расстояние от фишки до точки с координатами $(0, 0)$ больше 14 единиц. В начале игры фишка находится в точке с координатами $(3, S)$; $1 \leq S \leq 13$.

Ответьте на следующие вопросы:

Вопрос 1. Найдите минимальное значение S , при котором Петя не может выиграть за один ход, но Ваня выигрывает своим первым ходом после любого хода Пети.

Вопрос 2. Найдите два наименьших значения S , когда Петя имеет выигрышную стратегию, 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.

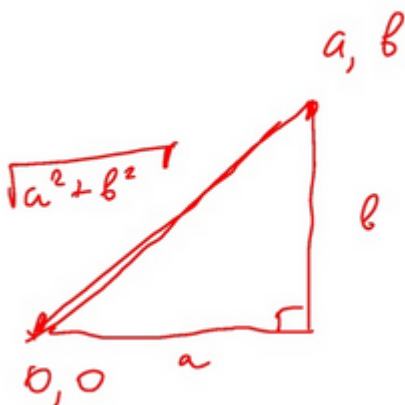
Вопрос 3. Найдите наибольшее значение S , при котором одновременно выполняются два условия:

- 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_456241300?t=1h46m40s

Решение

Решение этой задачи было нами уже выполнено аналитическим методом.



Используем для его решения нашу программу:

```
def f(a,b,m):
    if (a**2+b**2)**0.5>14: return m%2==0
    if m==0: return 0
    h = [f(2*a,b,m-1), f(a,b+3,m-1), f(a,b+4,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(1,14) if f(3,s,2)])
print(20,[s for s in range(1,14) if not f(3,s,1) and f(3,s,3)])
print(21,[s for s in range(1,14) if not f(3,s,2) and f(3,s,4)])
```

Результат работы программы:

19 [8, 9]

20 [4, 5, 6, 7]

21 [1, 2, 3]

Ответ:

- 1) 8
- 2) 4 5
- 3) 3

Задание №9(6254)

(PRO100 ЕГЭ) 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 три камня или одиннадцать камней. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится числом, оканчивающимся на ноль. Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, количество камней в которой оканчивается на ноль. К примеру, игра заканчивается, когда в куче стало 10, 200, 6800 камней. В начальный момент в куче было S камней, где S – двузначное число, не оканчивающиеся на ноль.

Ответьте на следующие вопросы:

Вопрос 1. Укажите минимальное значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.

Вопрос 2. Найдите количество значений 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.

Вопрос 3. Найдите сумму значений S , при которых у Вани есть выигрышная стратегия, причём 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_456241300?t=1h53m50s

Решение

Для ответа на третий вопрос задачи используем функцию `sum()`

```
def f(s,m):
    if s%10==0: return m%2==0
    if m==0: return 0
    h = [f(s+1,m-1),f(s+3,m-1),f(s+11,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(10,100) if s%10!=0 and f(s,2)])
print(20,[s for s in range(10,100) if s%10!=0 and not f(s,1) and f(s,3)])
print(21,sum([s for s in range(10,100) if s%10!=0 and not f(s,2) and f(s,4)]))
```

Результат работы программы:

19 [16, 26, 36, 46, 56, 66, 76, 86, 96]

20 [13, 15, 23, 25, 33, 35, 43, 45, 53, 55, 63, 65, 73, 75, 83, 85, 93, 95]

21 954

Ответ:

- 1) 16
- 2) 18
- 3) 954

Задание №10 (3487)

(А. Кабанов) Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в любую кучу один камень или увеличить количество камней в первой куче в два раза или увеличить количество камней во второй куче в три раза. Игра завершается в тот момент, когда общее количество камней в двух кучах становится не менее 30. Победителем считается игрок, сделавший последний ход. В начальный момент в первой куче было $K \geq 1$ камней, а во второй – $S \geq 1$ камней, $K + S \leq 29$.

Ответьте на следующие вопросы:

Вопрос 1. Сколько существует пар $(K; S)$, таких что Ваня выигрывает первым ходом при любой игре Пети?

Вопрос 2. При $S=7$, найдите минимальное и максимальное значение K , при котором 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 3. При $K=1$ найдите такое значение S , при котором, 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_456241300?t=1h58m25s

Решение

В этой задаче необходимо обратить внимание, на то, что ходы для разных куч различные. Поэтому важно, в каком порядке вы записываете первую и вторую кучу. Далее программа записывается аналогично программам в предыдущих задачах. В выводе на экран указываются условия записанные в вопросах.

```
def f(a,b,m):
    if a+b>=30: return m%2==0
    if m==0: return 0
    h = [f(a+1,b,m-1), f(a,b+1,m-1), f(a*2,b,m-1), f(a,b*3,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print(19, [(k,s) for k in range(1,30) for s in range(1,30) if k+s<=29 and f(k,s,2)])
print(20, [k for k in range(1,23) if not f(k,7,1) and f(k,7,3)])
print(21, [s for s in range(1,29) if not f(1,s,2) and f(1,s,4)])
```

Результат работы программы:

19 [(2, 9), (5, 8), (8, 7), (11, 6), (12, 5), (13, 3), (14, 1)]

20 [4, 5, 7]

21 [8]

Ответ :

1) 7

2) 4 7

3) 8

Telegram: @fast_ege