

. Текстовый разбор домашки 2

DZ_19-21_prog2_1

Задача №№1-3 (854)

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча, состоящая из S конфет. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может съесть не более пяти, но не менее одной конфеты или съесть половину конфет, если число конфет четное. Съесть можно только целое количество конфет. Игра завершается в тот момент, когда в куче останется менее десяти конфет. Победителем считается игрок, который сделал последний ход.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241305?t=0h0m0s

19. Укажите значение S , при котором Ваня выиграет первым ходом при любой игре Пети.
20. Укажите минимальное и максимальное S , при которых Петя не может выиграть первым ходом, но может выиграть вторым ходом при любом ходе Вани. Найденные значения запишите в ответе в порядке возрастания.
21. Укажите такое значение S , при котором 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.

Решение

Напишем рекурсивную функцию f от s, m . В ней определяем:

- Если значение s меньше 54, то возвращаем результат выражения $m \% 2 == 0$.
- Если $m == 0$, возвращаем 0.
- Далее опишем возможные ходы, определяем значения переменной h , которая зависит от шести вызовов функции:
 - о Первые пять вызовов: $f(s-1, m-1)$, $f(s-2, m-1)$, $f(s-3, m-1)$, $f(s-4, m-1)$, $f(s-5, m-1)$ – описываем ходы со съедением от 1 до 5 конфет)
 - о Шестой вызов: если s четное, то мы можем съесть половину конфет. Тогда h добавляется в ход $f(s \% 2, m-1)$.
- Далее возвращаем $\text{any}(h)$, если $(m-1) \% 2 == 0$, иначе — $\text{all}(h)$.

Для девятнадцатого задания нужно найти все значения s от 1 до 100 (берем число s запасом), для которых существует стратегия, позволяющая победить за два хода. Мы запускаем программу и получаем значение 15, сохраняем.

Для двадцатого задания: нужно найти два наименьших значения s , при которых Петя не может выиграть первым ходом, но может выиграть своим вторым ходом. Здесь важно, что Петя не выигрывает первым, но выигрывает вторым своим ходом (это третий ход игры), поэтому указываем условие $\text{if not } f(s, 1) \text{ and } f(s, 3)$. После запуска программы получаем список значений, из которых наименьшие — 17 и наибольшее 30.

Двадцать первое задание: нужно найти значение s , из которого Ваня может выиграть первым или вторым ходом, но не может гарантированно выиграть первым. Запускаем программу и получаем значение 21.

```
def f(s, m):  
    if s < 10: return m % 2 == 0
```

```

if m==0: return 0
h = [f(s-1,m-1), f(s-2,m-1), f(s-3,m-1), f(s-4,m-1), f(s-5,m-1)]
if s%2==0:
    h += [f(s//2,m-1)]
return any(h) if (m-1)%2==0 else all(h)

```

```

print(19,[s for s in range(10,100) if f(s,2)])
print(20,[s for s in range(10,100) if not f(s,1) and f(s,3)])
print(21,[s for s in range(10,100) if not f(s,2) and f(s,4)])

```

Ответ:

19. 15

20. 17 30

21. 21

Telegram: @fast_ege

DZ_19-21_prog2_2

Задача №№4-6 (8710)

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 один камень, либо, если в куче 4 или больше камней, он может убрать четыре камня, либо, если количество камней в куче кратно трем, он может уменьшить количество камней в куче в три раза.

Игра завершается в тот момент, когда количество камней в куче становится не более 1.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу из 1 камня или меньше.

В начальный момент в куче было S камней; $4 \leq S \leq 100$.

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

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241305?t=0h3m50s

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

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.

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.

Решение

Напишем рекурсивную функцию f от s , m . В ней определяем:

- Если значение s меньше или равно 1, то возвращаем результат выражения $m \% 2 == 0$.
- Если $m == 0$, возвращаем 0.
- В противном случае, определяем значения переменной h , которая зависит от вызовов функции f :
 - о Ход $s - 1$ мы можем выполнить всегда, то есть первый вызов: $f(s - 1, m - 1)$.
 - о Второй вызов: если s больше либо равен 4, то добавляем ход $f(s - 4, m - 1)$.
 - о Третий вызов: если s кратно трем, то добавляем ход $f(s // 3, m - 1)$.

Таким образом, у нас получается от 1 до 3 ходов.

Далее возвращаем `any(h)`, если $(m - 1) \% 2 == 0$, иначе — `all(h)`.

Для девятнадцатого задания нужно найти все значения s от 4 до 100, для которых существует стратегия, позволяющая победить за два хода. Мы запускаем программу и получаем значение 6.

Для двадцатого задания: нужно найти два наименьших значения s , при которых Петя не может выиграть первым ходом, но может выиграть своим вторым ходом. Петя не выигрывает первым, но выигрывает вторым своим ходом (это третий ход игры), поэтому указываем условие `if not f(s,1) and f(s,3)`. После запуска программы получаем значения 7, 10 и 18. Записываем в ответ 7 и 10.

Двадцать первое задание: нужно найти минимальное значение s , из которого Ваня может выиграть первым или вторым ходом, но не может гарантированно выиграть первым. Запускаем программу и получаем значения 8 и 11. Наименьшее значение — 8.

```
def f(s,m):
    if s<=1: return m%2==0
    if m==0: return 0
    h = [f(s-1,m-1)]
    if s>=4: h+=[f(s-4,m-1)]
    if s%3==0: h+=[f(s//3,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(4,101) if f(s,2)])
print(20,[s for s in range(4,101) if not f(s,1) and f(s,3)])
print(21,[s for s in range(4,101) if not f(s,2) and f(s,4)])
```

Ответ:

19. 6

20. 7 10

21. 8

Telegram: @fast_ege

DZ_19-21_prog2_3

Задача №№7-9 (4378)

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. Например, если в начале игры в куче 3 камня, Петя может первым ходом получить кучу из 4, 5 или 6 камней. Общий запас игроков составляет 50 камней (включая те, что уже лежат в куче).

Например, если в куче уже есть 30 камней, то следующим ходом выполнять удвоение нельзя — камней не хватит.

Игра завершается, когда количество камней в куче становится не менее 41. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 41 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 40$

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

Ссылка на видео-разбор с таймингом: https://vk.c om/video-205546952__456241305?t=0h7m30s

19. Укажите такое значение S , при котором 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.
20. Укажите два значения S , при которых у Вани есть выигрышная стратегия, позволяющая ему выиграть третьим ходом при любой игре Пети, но у Вани нет стратегии, которая позволяла бы ему гарантированно выиграть первым или вторым ходом. В ответе запишите найденные значения в порядке возрастания: сначала меньшее, затем большее.
21. Найдите такое значение S , при котором у Пети нет стратегии, позволяющей ему гарантированно выиграть первым ходом, но у Пети есть выигрышная стратегия, позволяющая ему выиграть вторым ходом при любой игре Вани, и при этом у Пети есть два разных первых хода, обеспечивающих выигрыш вторым ходом.

Решение

Напишем рекурсивную функцию f от s , m . В ней определяем:

- Если значение s больше или равно 41, то возвращаем результат выражения $m \% 2 == 0$.
- Если $m == 0$, возвращаем 0.
- Далее описываем возможные ходы в игре, определяем значения переменной h , которая зависит от разных вызовов функции:
 - о Первый вызов, ход можно сделать всегда: $f(s + 1, m - 1)$.
 - о Второй вызов, ход можно сделать всегда: $f(s + 2, m - 1)$.
 - о Третий вызов – добавляем ход только в случае, если в куче меньше 25 камней, только в этом случае можно удвоить количество: $f(s * 2, m - 1)$.
- Далее возвращаем $\text{any}(h)$, если $(m - 1) \% 2 == 0$, иначе — $\text{all}(h)$.

Для девятнадцатого задания нужно найти все значения s от 1 до 40, для которых существует стратегия, позволяющая победить за четыре, но не за два хода (второй ход Вани – это четвертый ход игры). Мы запускаем программу и получаем значение 35.

Для двадцатого задания: нужно найти два значения s , при которых Ваня не может выиграть первым ходом, не может выиграть своим вторым ходом, но может выиграть третьим ходом. (шестой ход игры) Указываем условие $\text{if not } f(s, 4) \text{ and } f(s, 6)$. После запуска программы получаем значения 17 и 32.

Двадцать первое задание: нужно найти значение s , из которого Петя может выиграть вторым ходом, но не может гарантированно выиграть первым. При этом у Пети есть два разных первых хода, обеспечивающих выигрыш вторым ходом. Запускаем программу для того, что б посмотреть, из каких значений можно выиграть вторым ходом, то есть для $f(s, 2)$. Получили значения 20, 38. Далее запустим программу и посмотрим, при каких значениях можно выиграть игру на третьем ходе, но нельзя выиграть на первом (то есть условие $\text{if not } f(s, 1) \text{ and } f(s, 3)$). Получили значения 10, 18, 19, 36, 37. Из этих значений подберем те, из которых сразу можно попасть и в 20, и в 38 (два разных хода, обеспечивающих победу Пете на втором ходе). Очевидно, это только число 19 (можно добавить 1 камень и получить 20, и удвоить количество камней, получить 38). 19 – ответ на задачу.

```
def f(s,m):
    if s>=41: return m%2==0
    if m==0: return 0
    h = [f(s+1,m-1),f(s+2,m-1)]
    if s<=25: h+=[f(s*2,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(1,41) if not f(s,2) and f(s,4)])
print(20,[s for s in range(1,41) if not f(s,4) and f(s,6)])
print([s for s in range(1,41) if f(s,2)])
```

```
#первый ход Пети попадает в B1
```

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

Ответ:

19. 35

20. 17 32

21. 19

Telegram: @fast_ege

DZ_19-21_prog2_4

Задача №№10-12 (5064)

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 не разрешается делать ход, после которого количество камней в куче будет делиться на 3. Например, если в начале игры в куче 4 камня, Петя может первым ходом получить кучу из 5 или из 8 камней. Добавить два камня Петя не может, так как в этом случае в куче станет 6 камней, а 6 делится на 3.

Игра завершается, когда количество камней в куче становится не менее 151. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 151 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 149$, S не делится на 3.

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

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241305?t=0h13m00s

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

20. Укажите два 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: сначала меньшее, затем большее.

21. Найдите такое 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 позволяла бы ему гарантированно выиграть первым ходом.

Решение

Напишем рекурсивную функцию f от s , m . В ней определяем:

- Если значение s больше или равно 151, то возвращаем результат выражения $m \% 2 == 0$.
- Если $m == 0$, возвращаем 0.
- Далее определяем значения переменной h . Для начала создадим пустой список $h=[]$, а дальше распишем возможные ходы:
 - о Если ход $+1$ попадает в число, не кратное трем $(s+1)\%3!=0$, то $h += f(s+1, m-1)$.
 - о Если ход $+2$ попадает в число, не кратное трем $(s+2)\%3!=0$, то $h += f(s+2, m-1)$.
 - о Если ход $*2$ попадает в число, не кратное трем $(s*2)\%3!=0$, то $h += f(s*2, m-1)$.
- Далее возвращаем $\text{any}(h)$, если $(m-1) \% 2 == 0$, иначе — $\text{all}(h)$.

Для девятнадцатого задания нужно найти все значения s от 1 до 149, для которых существует стратегия, позволяющая победить за два хода, и при этом s не делится на 3. Мы запускаем программу и получаем значение 74. Указываем его в ответе на задачу.

Для двадцатого задания: нужно найти наименьшее и наибольшее значения s , при которых Петя не может выиграть первым ходом, но может выиграть своим вторым ходом (это третий ход игры), поэтому указываем условие `if not f(s,1) and f(s,3)`, и при этом s не делится на 3. После запуска программы получаем два значения, из которых минимальное 37 и максимальное - 73.

Двадцать первое задание: нужно найти значение s , из которого Ваня может выиграть первым или вторым ходом, но не может гарантированно выиграть первым, и при этом s не делится на 3. Запускаем программу и получаем значение 71.

```
def f(s,m):
    if s>=151: return m%2==0
    if m==0: return 0
    h = []
    if (s+1)%3!=0: h+= [f(s+1,m-1)]
    if (s+2)%3!=0: h+= [f(s+2,m-1)]
    if (s*2)%3!=0: h+= [f(s*2,m-1)]
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(1,150) if s%3!=0 and f(s,2)])
print(20,[s for s in range(1,150) if s%3!=0 and not f(s,1) and f(s,3)])
print(21,[s for s in range(1,150) if s%3!=0 and not f(s,2) and f(s,4)])
```

Ответ:

19. 74

20. 37 73

21. 71

Telegram: @fast_ege

DZ_19-21_prog2_5

Задача №№13-15 (2575)

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 три раза. При этом нельзя повторять ход, который только что сделал второй игрок. Например, если в начале игры в куче 4 камня, Петя может первым ходом получить кучу из 5, 6 или 12 камней. Если Петя добавил 1 камень и получил кучу из 5 камней, то следующим ходом Ваня может либо добавить 2 камня (и получить 7 камней), либо утроить количество камней в куче (их станет 15). Получить 6 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 62. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 62 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 61$.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241305?t=0h17m35s

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.

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.

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.

Решение

Напишем рекурсивную функцию f от s , m и p . p – это предыдущий выполненный ход. В ней определяем:

- Если значение s больше или равно 62, то возвращаем результат выражения $m \% 2 == 0$.
- Если $m == 0$, возвращаем 0.
- Далее определяем значения переменной h . Для начала создадим пустой список $h=[]$, а дальше распишем возможные ходы:
 - о Если предыдущий ход p не был добавлением одного камня, то есть $p != '+1'$, то $h += f(s+1, m-1, '+1')$, значение p меняется на $' +1'$, следующий игрок будет знать о том, что это был за ход.
 - о Если предыдущий ход p не был добавлением двух камней, то есть $p != '+2'$, то $h += f(s+2, m-1, '+2')$, значение p меняется на $' +2'$.
 - о Если предыдущий ход p не был увеличением кучи камней в три раза, то есть $p != '*3'$, то $h += f(s*3, m-1, '*3')$, значение p меняется на $' *3'$.
- Далее возвращаем $any(h)$, если $(m-1) \% 2 == 0$, иначе — $all(h)$.

Для девятнадцатого задания нужно найти значение s от 1 до 61, для которых существует стратегия, позволяющая победить за два хода. Мы запускаем программу (указав стартовое значение $p = ''$ – пустая строка, любой начальный ход) значение – 20, указываем его в ответе на задачу.

Для двадцатого задания нужно найти наименьшее и наибольшее значения s , при которых Петя не может выиграть первым ходом, но может выиграть своим вторым ходом. Это третий ход игры, поэтому указываем условие $if not f(s,1, '') and f(s,3, '')$, $p = ''$ – пустая строка, любой начальный ход. После запуска программы получаем список всех значений от 7 до 19, из которых минимальное 7 и максимальное – 19.

Двадцать первое задание: нужно найти значение s , из которого Ваня может выиграть первым или вторым ходом, но не может гарантированно выиграть первым: указываем условие $if not f(s,2, '') and f(s,4, '')$, $p = ''$ – пустая строка, любой начальный ход. Запускаем программу и получаем значение 6.

```
def f(s,m,p):
    if s>=62: return m%2==0
    if m==0: return 0
    h = []
    if p!='+1': h+=f(s+1,m-1,'+1')
    if p!='+2': h+=f(s+2,m-1,'+2')
    if p!='*3': h+=f(s*3,m-1,'*3')
    return any(h) if (m-1)%2==0 else all(h)

print(19,[s for s in range(1,62) if f(s,2,'')])
print(20,[s for s in range(1,62) if not f(s,1,'') and f(s,3,'')])
print(21,[s for s in range(1,62) if not f(s,2,'') and f(s,4,'')])
```

Ответ:

19. 20

20. 7 19

21. 6

Telegram: @fast_ege

DZ_19-21_prog2_6

Задание №16-18 (4374)

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 21. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 21 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 20$.

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

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241305?t=0h21m45s

19. Укажите наименьшее значение S , при котором Петя не может выиграть за один ход, но у Пети есть выигрышная стратегия, позволяющая ему выиграть вторым ходом.

20. 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.

21. 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.

Решение

Напишем рекурсивную функцию f от s , m и tr . tr – это траектория, список ходов игроков. В ней определяем:

- Если значение s больше или равно 62, то возвращаем результат выражения $m \% 2 == 0$.
- Если $m == 0$, возвращаем 0.
- Далее определяем значения переменной h . Для начала создадим пустой список $h = []$, а дальше распишем возможные ходы:
 - о Если наш предыдущий ход $tr[-2]$ не равен единице, то есть $tr[-2] \neq '1'$, то $h += f(s + 1, m - 1, tr + '1')$, к траектории добавляется единица, единицей назовем плюс один ход.
 - о Если наш предыдущий ход $tr[-2]$ не равен двум, то есть $tr[-2] \neq '2'$, то $h += f(s + 2, m - 1, tr + '1')$, к траектории добавляется два, какой-то ход под номером 2.
 - о Если наш предыдущий ход $tr[-2]$ не равен трем, то есть $tr[-2] \neq '3'$, то $h += f(s * 2, m - 1, tr + '1')$, к траектории добавляется три, так собирается траектория ходов.
- Далее возвращаем $\max(h)$, если $(m - 1) \% 2 == 0$, иначе — $\min(h)$.

Для девятнадцатого задания нужно найти наименьшее значение s от 1 до 20, для которого существует стратегия, позволяющая победить Пете вторым ходом. При этом пишем стартовую строку так, чтобы корректно работал обратный индекс поэтому в качестве третьего элемента функции f мы напишем 00, это просто символ пустышка, которого нет в условии, чтобы в первый ход Пети и первый ход Вани были все возможные варианты ходов. Запускаем программу и получаем значения 8 и 9, указываем 8 в ответе на задачу.

Для двадцатого задания нужно указать наименьшее и наибольшее значения s , при которых Ваня не может выиграть первым ходом, но может выиграть своим вторым ходом, это четвертый ход игры. Начальную траекторию опять же указываем 00. После запуска программы получаем значения 6 и 7.

Двадцать первое задание: найти наибольшее значение S , при котором у Пети есть выигрышная стратегия, позволяющая ему выиграть третьим ходом (это пятый ход игры). Запускаем программу для условий `if not f(s,3,'00') and f(s,5,'00')` и получаем значения 3 и 5. Записываем в ответ 5.

```
def f(s,m,tr):
    if s>=21: return m%2==0
    if m==0: return 0
    h = []
    if tr[-2]!='1': h+= [f(s+1,m-1,tr+'1')]
    if tr[-2]!='2': h+= [f(s+2,m-1,tr+'2')]
    if tr[-2]!='3': h+= [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(1,21) if not f(s,1,'00') and f(s,3,'00')])
print(20,[s for s in range(1,21) if not f(s,2,'00') and f(s,4,'00')])
print(21,[s for s in range(1,21) if not f(s,3,'00') and f(s,5,'00')])
```

Ответ:

19. 8

20. 6 7

21. 5

Telegram: @fast_ege

DZ_19-21_prog2_7

Задание №19-21(727)

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча, состоящая из S конфет. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может съесть не более половины от всех оставшихся конфет, но не менее одной конфеты. Игра завершается в тот момент, когда в куче не остается ни одной конфеты. Победителем считается игрок, который съел последнюю конфету.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241305?t=0h27m25s

19. Укажите максимальное двузначное значение S , при котором у Вани есть выигрышная стратегия.

20. Укажите минимальное трехзначное значение S , при котором у Пети есть выигрышная стратегия.

21. Сколько существует трехзначных значений S , при которых выигрышная стратегия есть у Вани?

Решение

Подключим `functools` для ускорения работы программы, потому что дерево ходов получается очень большим, пишем `from functools import *`. Добавим кэш, пишем `@lru_cache(None)`.

Далее напишем рекурсивную функцию f от s , m . В ней определяем:

- Если значение s равно 0, то возвращаем результат выражения $m \% 2 == 0$.
- Если $m == 0$, возвращаем 0.
- Далее опишем возможные ходы, определяем значения переменной h :
 - о Всегда есть ход «съесть одну конфету», т.е. $f(s-1, m-1)$.
 - о Еще ход $f(s-i, m-1)$ for i in $\text{range}(1, s//2+1)$ – то есть можно съесть до половины всех конфет. Ход «съесть одну конфету» мы описываем отдельно для того, чтобы цикл работал корректно.
- Далее возвращаем $\text{any}(h)$, если $(m-1) \% 2 == 0$, иначе — $\text{all}(h)$.

Для девятнадцатого задания нужно найти максимальное двузначное значение S , то есть s от 10 до 99, при котором у Вани есть выигрышная стратегия. Указываем условие $f(s, 100)$ – с запасом, пусть у Вани будет до 100 ходов. Запускаем программу и получаем список значений. Максимальное двузначное значение – 95, указываем его в ответе на задачу.

Для двадцатого задания нужно найти минимальное трехзначное значение S , при котором у Пети есть выигрышная стратегия., поэтому указываем условие s от 100 до 1000 и условие $\text{if } f(s, 101)$ – с запасом, число нечетное. После запуска программы получаем значения, из которых минимальное трехзначное равно 100.

Двадцать первое задание: нужно найти сколько существует трехзначных значений S , при которых выигрышная стратегия есть у Вани. Запускаем программу (указываем s от 100 до 1000 и условие $f(s, 100)$) и получаем список значений из трех штук. Записываем в ответ 3.

```
from functools import *

@lru_cache(None)
def f(s, m):
    if s==0: return m%2==0
    if m==0: return 0
    h = [f(s-1, m-1)] + [f(s-i, m-1) for i in range(1, s//2+1)]
    return any(h) if (m-1)%2==0 else all(h)

print(19, [s for s in range(10, 100) if f(s, 100)])
print(20, [s for s in range(100, 1000) if f(s, 101)])
print(21, [s for s in range(100, 1000) if f(s, 100)])
```

Ответ:

19. 95

20. 100

21. 3

Telegram: @fast_ege

DZ_19-21_prog2_8

Задание №22-24(471)

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней, количество которых известно. Первый ход делает Петя, игроки ходят по очереди один за другим. Игрок может либо увеличить количество камней на 2, либо увеличить количество втрое. Игра завершается, когда игрок приходит в кучу из N или более камней.

Так, например, если перед ходом игрока была куча из 20 камней, то своим ходом он может сделать кучу из 22 или 60 камней.

Известно, что в начале игры в куче было 15 камней.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241305?t=0h32m25s

19. Найдите все значения N , в которых должна закончиться игра, если известно, что Ваня имеет выигрышную стратегию своим первым ходом, то есть может выиграть при любой игре Пети. В качестве ответа укажите наименьшее и наибольшее конечные значения.

20. Известно, что при игре из кучи, в которой было 10 камней, Петя имеет выигрышную стратегию своим вторым ходом. Укажите минимальное и максимальное значения N , при которых это возможно.

21. Известно, Ваня выигрывает своим первым или вторым ходом при любой игре Пети, при этом Ваня не имеет гарантированной стратегии победы своим первым ходом. Игра ведется из кучи, в которой 5 камней. Сколько существует значений N , при которых это возможно?

Решение

Напишем рекурсивную функцию f от s , m и end . end – это условие окончания. В ней определяем:

- Если значение s больше или равно end , то возвращаем результат выражения $m \% 2 == 0$.
- Если $m == 0$, возвращаем 0.
- Далее определяем значения переменной h существует всего два возможных хода:
 - o $f(s + 2, m - 1, end)$
 - o $f(s * 3, m - 1, end)$
- Далее возвращаем $any(h)$, если $(m - 1) \% 2 == 0$, иначе — $all(h)$.

Для девятнадцатого задания нужно найти все значения N , в которых должна закончиться игра, если известно, что Ваня имеет выигрышную стратегию своим первым ходом, то есть может выиграть при любой игре Пети. Мы запускаем программу (указав значение end от 16 (минимальное количество камней равно 15) до 100, и условие $f(15, 2, end)$ и получаем значения от 46 до 51, указываем в ответе на задачу 46, 51.

Для двадцатого задания известно, что в куче было 10 камней и нужно найти наименьшее и наибольшее значения N , при которых Петя имеет выигрышную стратегию вторым ходом. Будем перебирать значения end от 11 до 100, условие выигрыша не первым, но третьим ходом в игре - $if not f(10, 1, end) and f(10, 3, end)$. После запуска программы получаем список значений от 37 до 96, из который минимальное 37 и максимальное - 96.

Двадцать первое задание: игра ведется из кучи в 3 камня, нужно найти количество значений N при которых Ваня может выиграть первым или вторым ходом, но не может гарантированно выиграть первым. Будем перебирать значения end от 6 до 100, условие выигрыша не вторым, но четвертым ходом в игре - $if not f(5, 2, end) and f(5, 4, end)$. После запуска программы получаем список значений от 28 до 33, всего 6 штук.

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

print(19, [end for end in range(16, 100) if f(15, 2, end)])
print(20, [end for end in range(11, 100) if not f(10, 1, end) and f(10, 3, end)])
print(21, [end for end in range(6, 100) if not f(5, 2, end) and f(5, 4, end)])
```

Ответ:

19. 49 51

20. 37 96

21. 6

Telegram: @fast_ege