Смысл игровых стратегий, как и любых других стратегий, заключается в том, чтобы приблизиться к цели как можно более эффективно. Либо проще и более глобально – **одержать победу** над противником.

Игроков, которые стараются победить, а не делают ходы наугад, мы тоже будем называть **разумными**.

Каждой **игре с полной информацией** (термин теории игр, обозначающий логическую игру, в которой для соперников отсутствует элемент неопределённости) , правила которой не допускают ничьей, существует **выигрышная стратегия** для одного из игроков. Выигрышная стратегия —это правило, следуя которому, один из игроков может выиграть, как бы ни играл его противник. Для каждой игры с определёнными правилами **выигрышную стратегию может иметь только один из игроков**.



Для того чтобы найти **выигрышную стратегию,** нужно последовательно рассмотреть все возможные позиции игры. Все позиции игры можно перебрать, **построив дерево игры (граф)**. Но для некоторых игр удаётся построить выигрышную стратегию без перебора всех возможных позиций. Одна из немногих игр, в которых можно перебрать все позиции, не строя дерева игры —это игра Камешки.

Понятие стратегии не надо отождествлять с понятием хода.**Стратегия определяет полный план действий игрока при всевозможных ситуациях, могущих возникнуть в игре.**

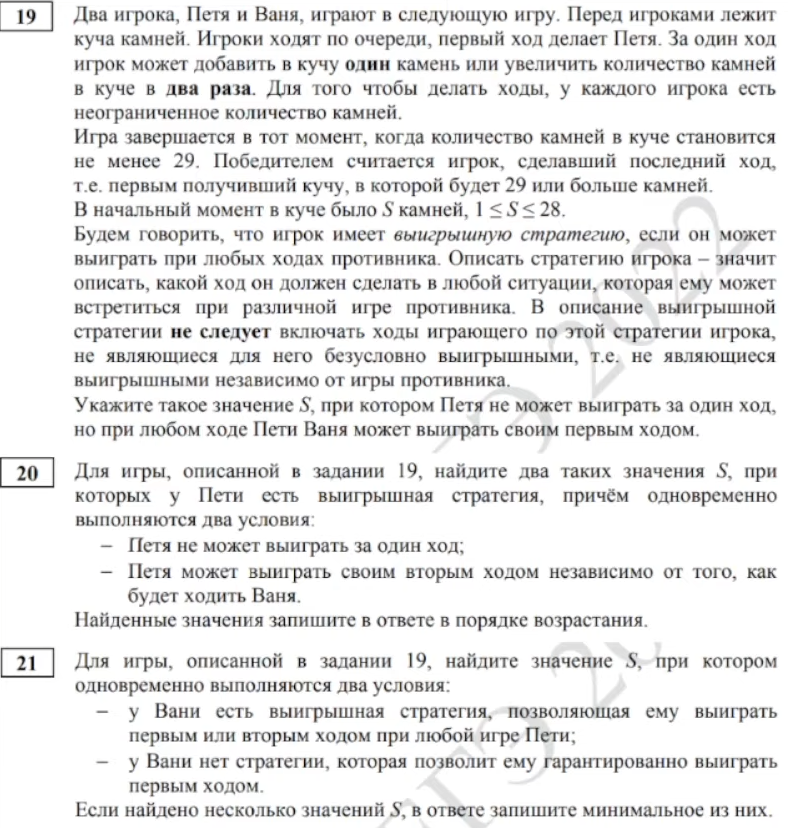
**Позиция выигрышная**, если из неё есть ход, который оставит противнику проигрышную позицию.

**Позиция проигрышная**, если любой ход из неё оставляет противнику выигрышную позицию.

Результат многих логических игр, в том числе и шахмат, предсказуем. Почему же мы не только не знаем выигрышную стратегию для большинства популярных игр, но открытым остается и вопрос, кто имеет гарантированный выигрыш в каждой из них? Дело здесь в астрономическом числе возможных позиций. Дерево игры в шахматы огромно: на первом уровне такого дерева 20 вершин, на втором —уже 400вершин!  Полное число возможных позиций в шахматах порядка 10120!



КАК РЕШИТЬ ЭТО РУКАМИ?

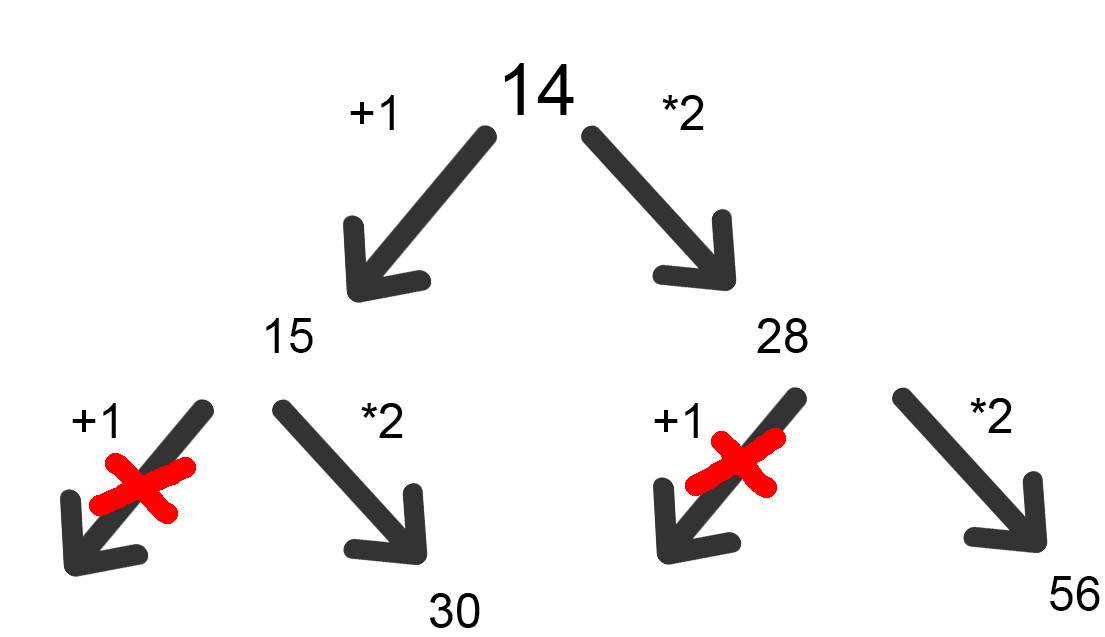


19.

**Ходы: +1; \*2**

**Игра закончиться при S> =29 камней.**

Давайте для начала возьмем **S = 14**. Что мы сможем сделать c начальным количеством? В какие позиции мы попадем?



При таком значение S, можно расписать всевозможные исходы игры. Очевидно, что первым ходом Петя **не** выиграет своим ходом, а своим первым ходом Ваня одержит победу (при любой игре Пети).

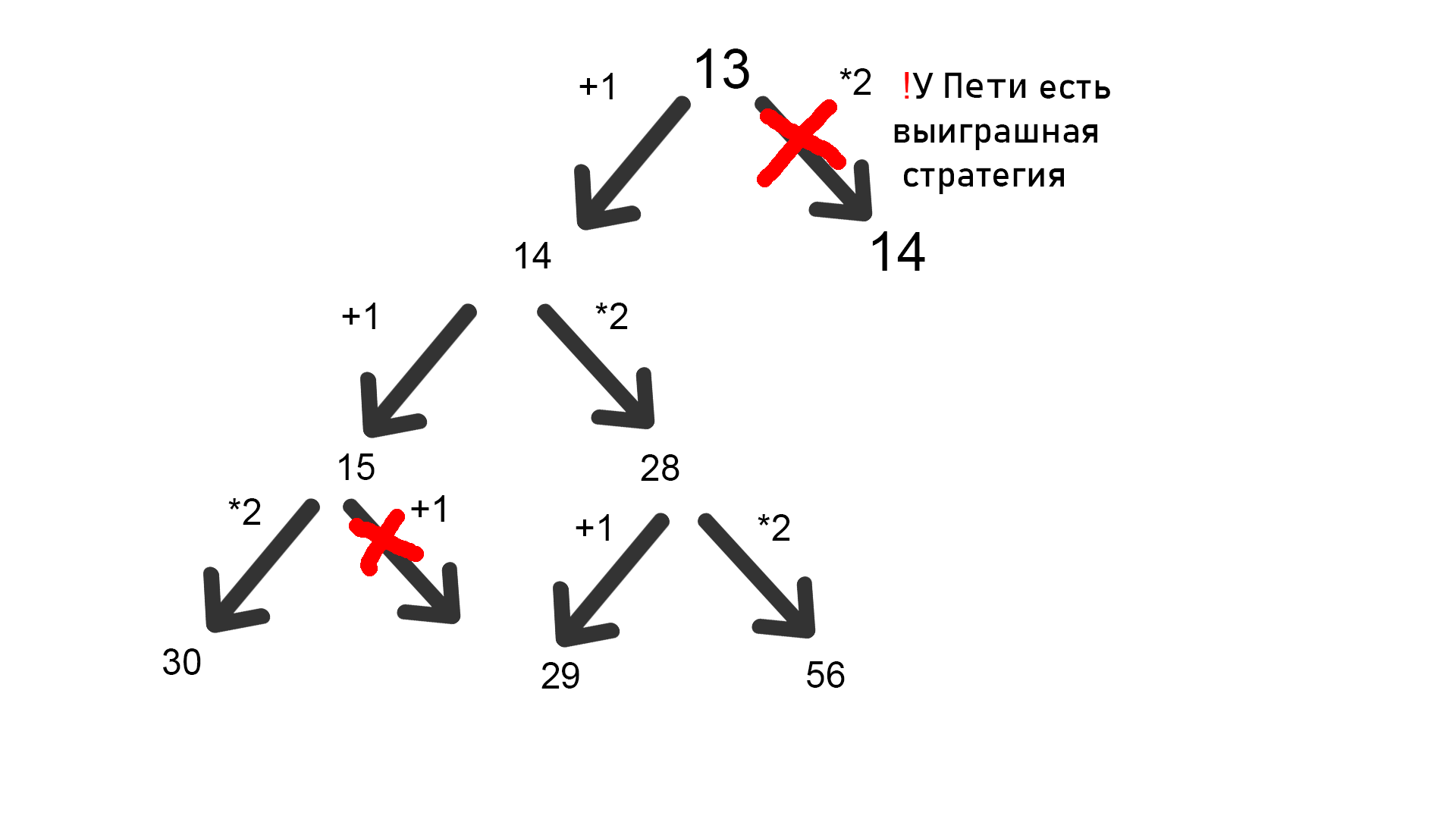
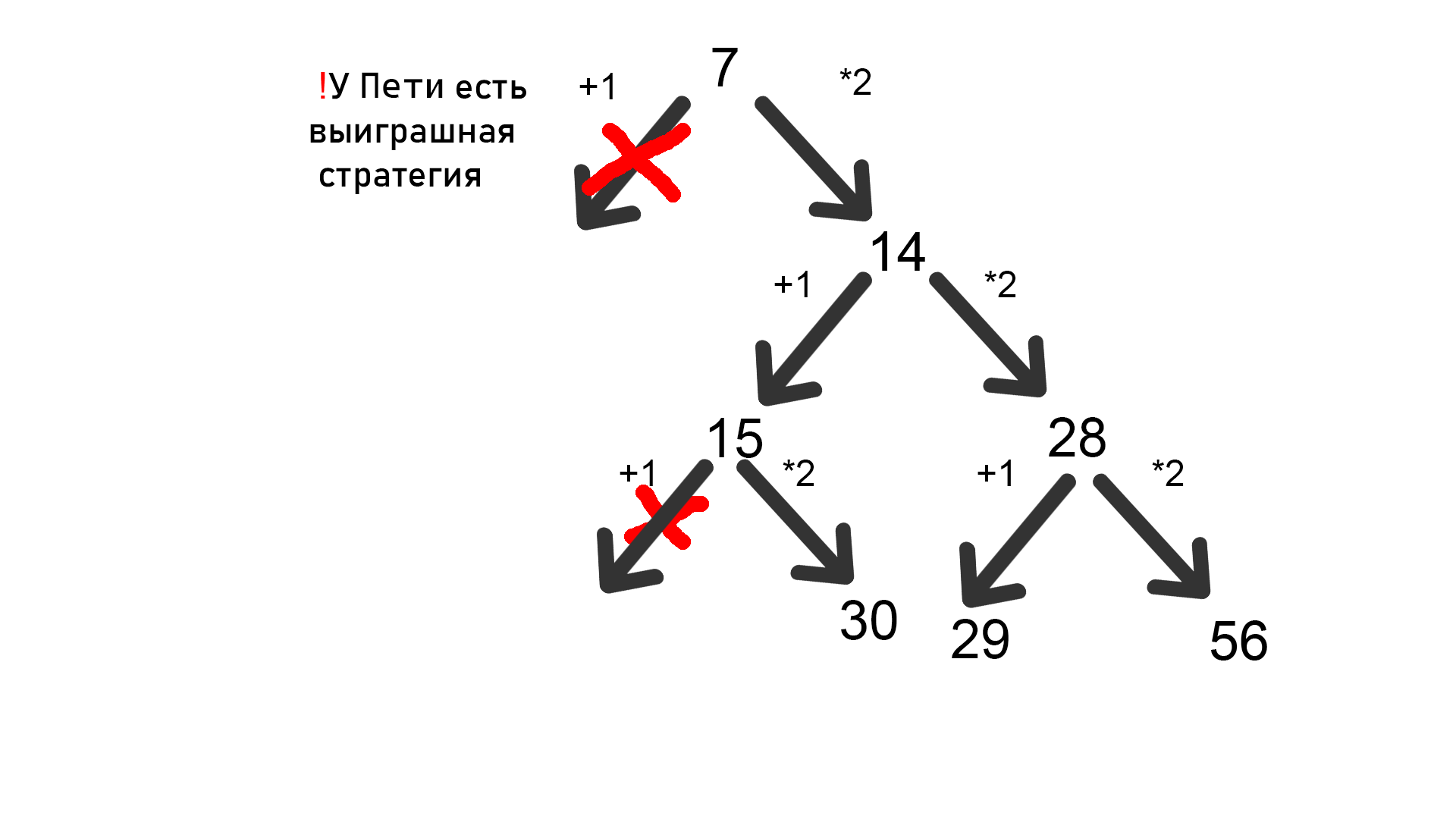
ОТВЕТ: 14.

20.

**Ходы: +1; \*2**

**Игра закончиться при S> =29 камней.**

**Нам нужно, чтобы после хода Пети, в куче осталось 14, так как из предыдущего задания мы знаем, что эта позиция будет выигрышной. Легко заметить, что в 14 мы можем попасть из 7 и 13(так как ходы +1 и \*2)**



ОТВЕТ: 13;7.

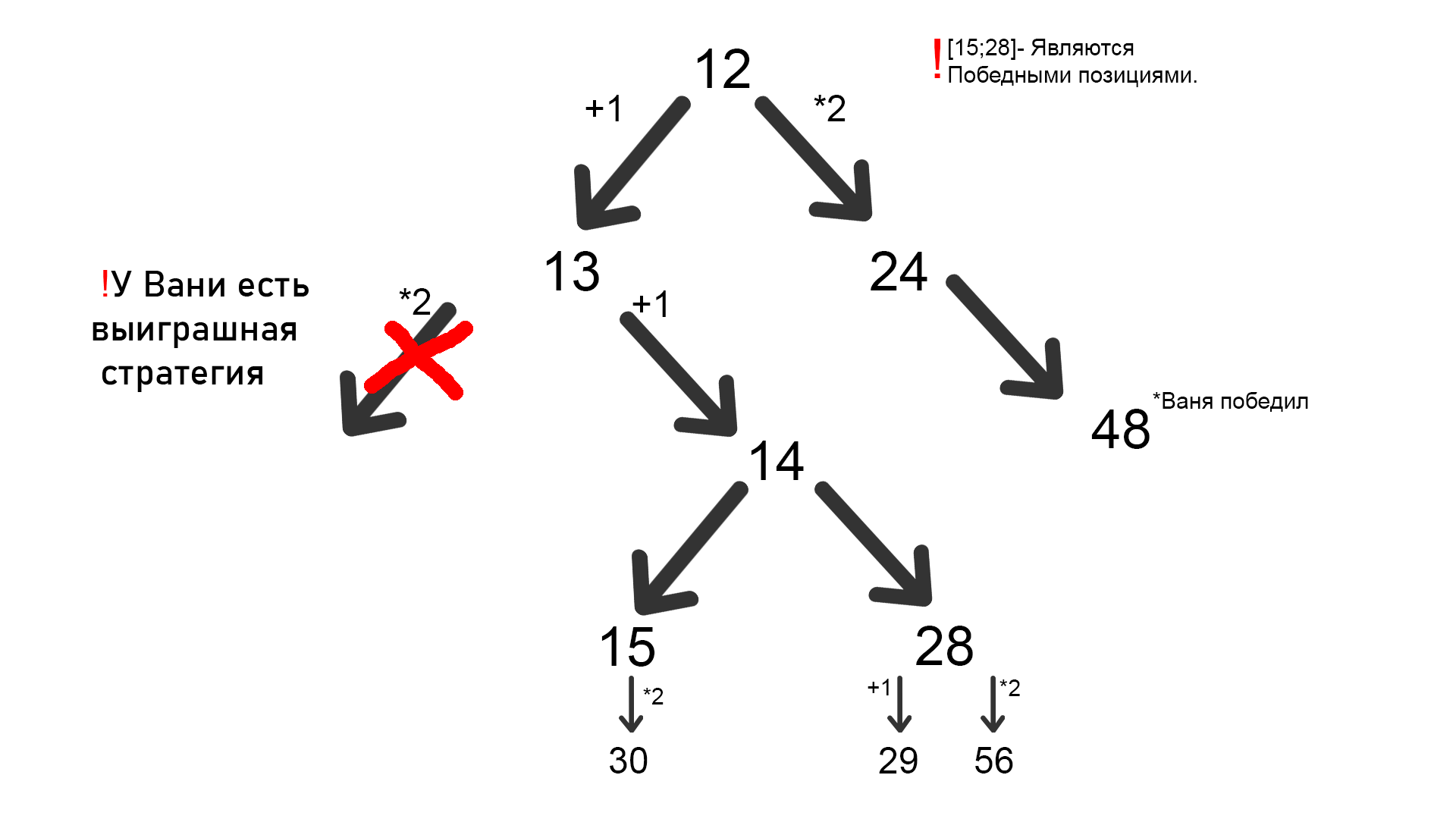
21.

**Ходы: +1; \*2**

**Игра закончиться при S> =29 камней.**

Очевидно, что, взяв изначальное число 12, Петя не сможет выиграть своим первым ходом,

А также он не сможет выиграть своим вторым ходом, поскольку не будет хватать камней.



Ответ 12.

РАЦИОНАЛЬНЫЙ СПОСОБЫ РЕШЕНИЯ ЗАДАЧ 19-21 ЕГЭ.

Excel

**P-00**(**демо-2021**).Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза**. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее **77**. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, при которой в кучах будет 77 или больше камней. В начальный момент в первой куче было **семь** камней, во второй куче – S камней; **1 ≤ S ≤ 69**.

**Задание 19.**

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

**Задание 20.**

Найдите два таких значения S, при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

− Петя не может выиграть за один ход;

− Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

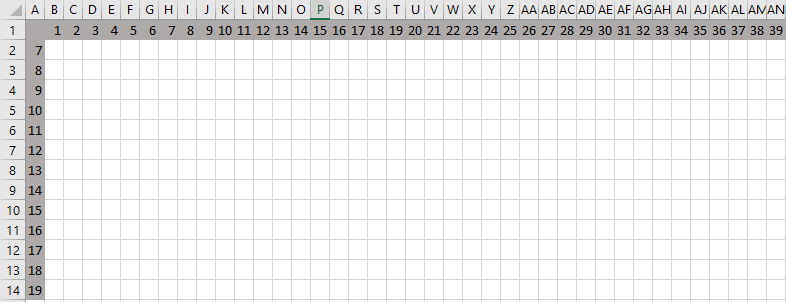
Найденные значения запишите в ответе в порядке возрастания.

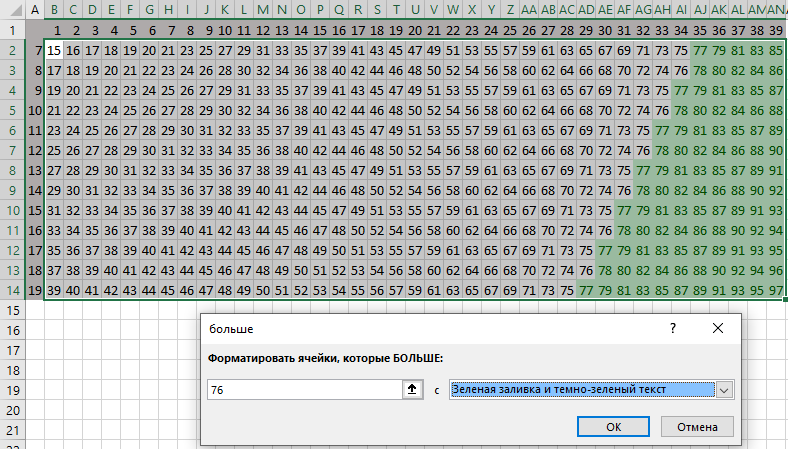
**Задание 21**

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.

1. Для начала отметим комбинации камней, из которых игрок побеждает своим первым ходом. Составим таблицу, в которой по вертикали отметим камни в первой куче камней (начиная с 7), а по горизонтали – во второй куче камней (начиная с 1).
2. Для каждой пары рассчитаем максимальное число камней, которое может получить игрок за один ход. Для этого необходимо большую из куч умножить на два. В ячейке B2 запишем формулу =МАКС(2\*$A2+B$1;$A2+2\*B$1) и заполним ей всю нашу таблицу. Далее с помощью условного форматирования пометим ячейки, в которых сумма не менее 77 (условие выигрыша).



**Решение 19 задания:**

1. Для решения 19 задания рассмотрим две ситуации:

а) Петя умножает вторую кучу на 2, а Ваня выигрывает первым ходом. Ход Пети имеет вид (7;S) →(7;2S). Ваня будет выигрывать первым ходом, если 2S ≥35 (см. таблицу) или S≥18.

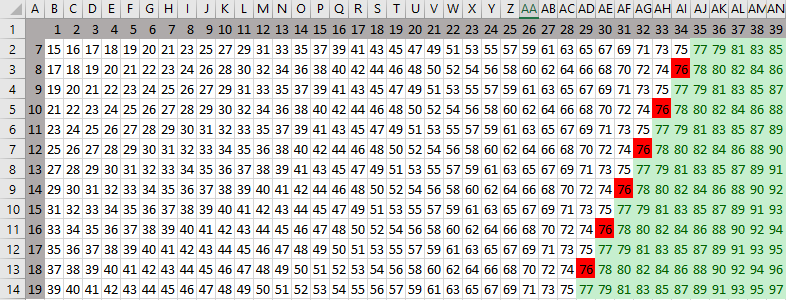
б) Петя умножает первую кучу на 2, а Ваня выигрывает первым ходом. Ход Пети имеет вид (7;S) →(14;S). Ваня будет выигрывать первым ходом, если S ≥32 (см. таблицу).

1. Нетрудно видеть, что минимальное подходящее значение S равно 18.

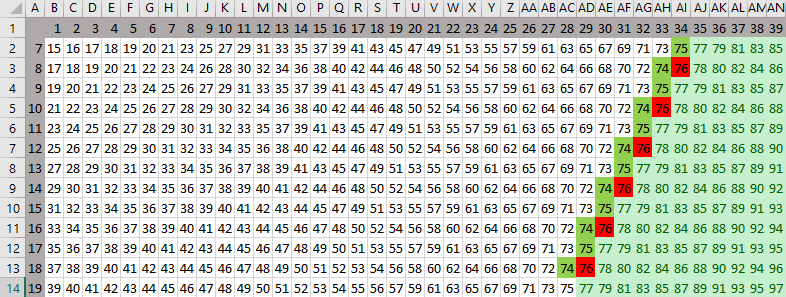
Ответ: 18.

**Решение 20 задания:**

1. Для начала найдём проигрышные значения (при любой игре побеждает следующий игрок). Рассмотрим комбинации (K; S) из которых каждый ход вида (K; S+1) (K+1; S) (2K; S) (K; 2S) попадает в выигрышные позиции.



1. Петя будет гарантированно побеждать своим вторым ходом, если его первый ход попадает в отмеченные проигрышные значения.
2. Рассмотрим ситуации, когда победный ход Пети (K+1;S) или (K;S+1). Отметим в таблице ячейки, из которых возможны такие ходы.



1. Рассмотрим ситуации, когда победный ход Пети (2K; S) или (K;2S). Выпишем все проигрышные позиции:

(8;34) (10;33) (12;32) (14;31) (16;30) (18;29) …

1. Для каждой из них выпишем, позиции с вдвое меньшим числом камней в одной из куч (если такая позиция возможна):

(8;34) – (8;17)

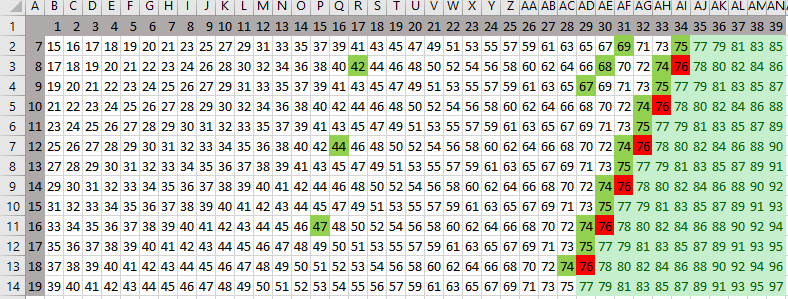
(12;32) – (12;16)

(14;31) – (7;31)

(16;30) – (8;30) (16;15)

(18;29) – (9;29)

1. Также отметим их на таблице (остальные по необходимости добавляются по аналогии)



1. Интерес представляют значения, где первая куча равна 7, поэтому:

Ответ: 31 34.

**Решение 21 задания:**

1. Рассмотрим комбинации вида (7; S), ходы (K+1;S) и (K;S+1) из которых попадает в выигрышные позиции. По таблице видно, что это (7;33) и (7;30). Проверим их:

– Из (7;33) Петя может сходить в (7;34) (8;33) – победа Вани 2 ходом и (7;66) (14;33) – победа Вани 1 ходом.

– Из (7;30) Петя может сходить в (7;31) (8;30) (14;30) – победа Вани 2 ходом и (7;62) – победа Вани 1 ходом.

1. Оба значения подходят и наименьшее из этих значений 30.

Ответ: 30.

C++

**Перед этим разбором мы настоятельно рекомендуем ознакомиться с рекурсией.**

cppstudio.com/post/418

vector<int>v(2000,INT\_MIN); // создаем вектор, который будет вмещать в себя все “ходы” игроков.

//Инициализируем все значения(2000) вектора INT\_MIN.

int game(int x){

if(v[x] != INT\_MIN){ // возвращаем возможные “ходы”.

return v[x];

}

if(x >= 1000){ //Очевидно, что если число больше 1000, то мы должны вернуть 0.

return 0;

}

vector<int>neg,tmp={game(x+100),game(x\*2)}; // определяем рекурсивно все возможные “ходы”.

for(int x: tmp){ // “кладем ходы” в вектор.

if(x<=0){

neg.push\_back(x);

}

}

*// основной код функции*

int res; **// результат**

if(!neg.empty()){ **// если вектор neg не пустой, то мы должны положить в результат**

res = -\*max\_element(neg.begin(),neg.end())+1;

**// Берем максимальное из отрицательных значений или 0, т.к. Наличие отрицательного(или 0) числа говорит нам о том, что в этой комбинации есть победные значение , чтобы сходить в эту позицию, нам нужно вернуть число по модулю и добавить к нему 1.**

} else {

res = -\*max\_element(tmp.begin(),tmp.end());

**// если в векторе tmp нет отрицательных, то игрок в позиции x и y проигрывать и его цель затянуть игру, то есть его цель – проиграть за максимальное количество шагов.**

}

return v[x] = res; **// возвращаем элемент**

}

Пример:

Позиция x y

Tmp = {1,**-1**, 2, 1} // вектор содержит отрицательные значения -1, это означает, что игрок, делающий ход из позиции x y, победит, потому что он будет ходить в -1 , тем самым заведет второго игрока в проигрышную позицию.

// значит, что функция вернет 2, и значит, что в позиции x y игрок выигрывает, сделав максимум 2 ходы

Tmp = {1, **1**, 2, 1}} // вектор не содержит отрицательные значения, это означает, что игрок, делающий ход из позиции x y, проиграет, его цель – как можно дальше отнять проигрыш, поэтому мы берем максимальное значение.

// значит, что функция вернет -2. И значит, что в позиции x y игрок проиграет, сделав максимум 2 ход

Дальше просто перебираем циклом все значения в основной программе

int main()

{

//int c =0;

for(int i = 1; i < 100; i++){

if (game(2,i,2 \* i) == 2){

cout << "(" << 6 << " , " << i <<") " << game(2,i,2 \* i) << endl;

}

}

}

САМЫЕ РАСПРОСТРАНЕННЫЕ СПОСОБЫ РЕШЕНИЯ ЗАДАЧ 19-21 ЕГЭ.

Excel

ЗАДАЧА (ПРИМЕР) САЙТ ПОЛЯКОВА(https://kpolyakov.spb.ru)

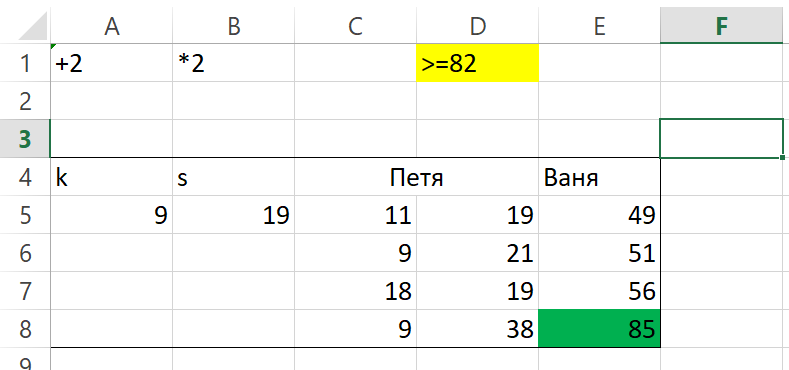
|  |
| --- |
| Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в одну из куч **два камня** или **увеличить** количество камней в куче **в два раза**. Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 82. Победителем считается игрок, сделавший последний ход, т. е. первым получивший позицию, в которой в кучах будет 82 или больше камней. В начальный момент в первой куче было 9 камней, во второй куче – S камней, 1 ≤ S ≤ 72. 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. |

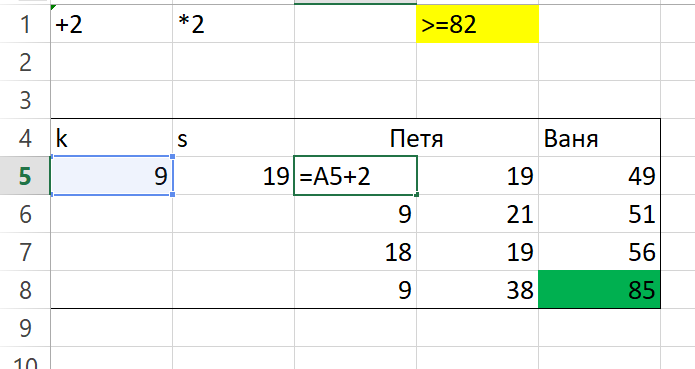
19. **Ходы: +2; \*2**

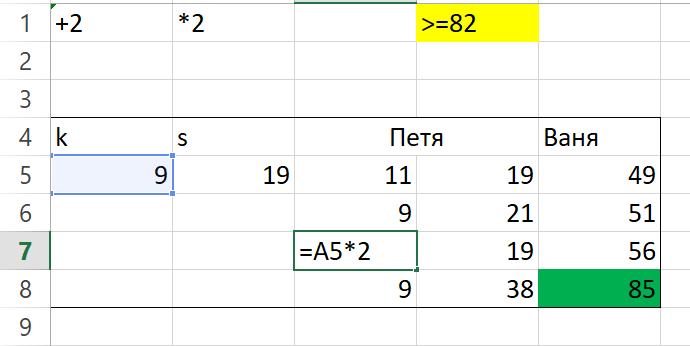
**Игра закончиться при S> =82 камней.**

1) Обозначим для себя условие.

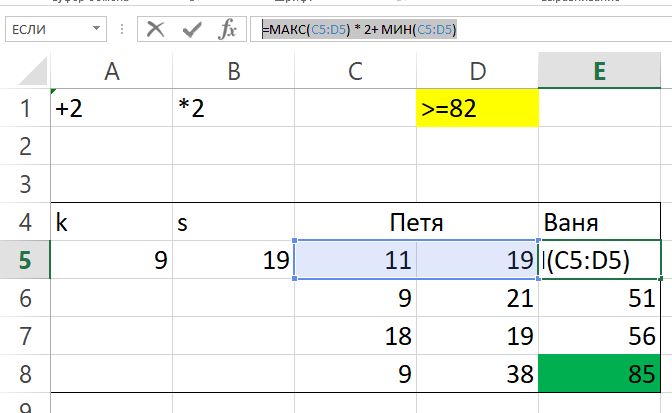
2) Обозначим кучки K и S (К по условию задачи = 9)



1. Применяем условие к кучам.

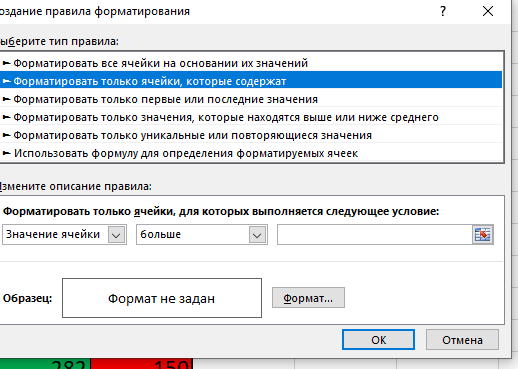
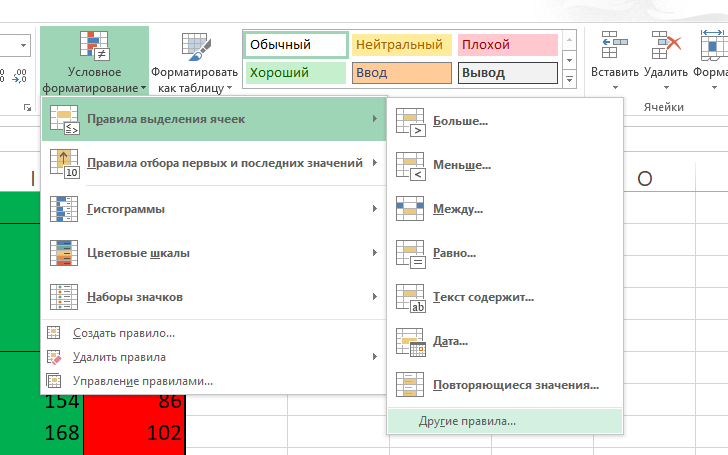


1. Так мы должны складывать кучи в ячейках у нас должна быть следующая формула:



“=МАКС(C5:D5) \* 2+ МИН(C5:D5)”.

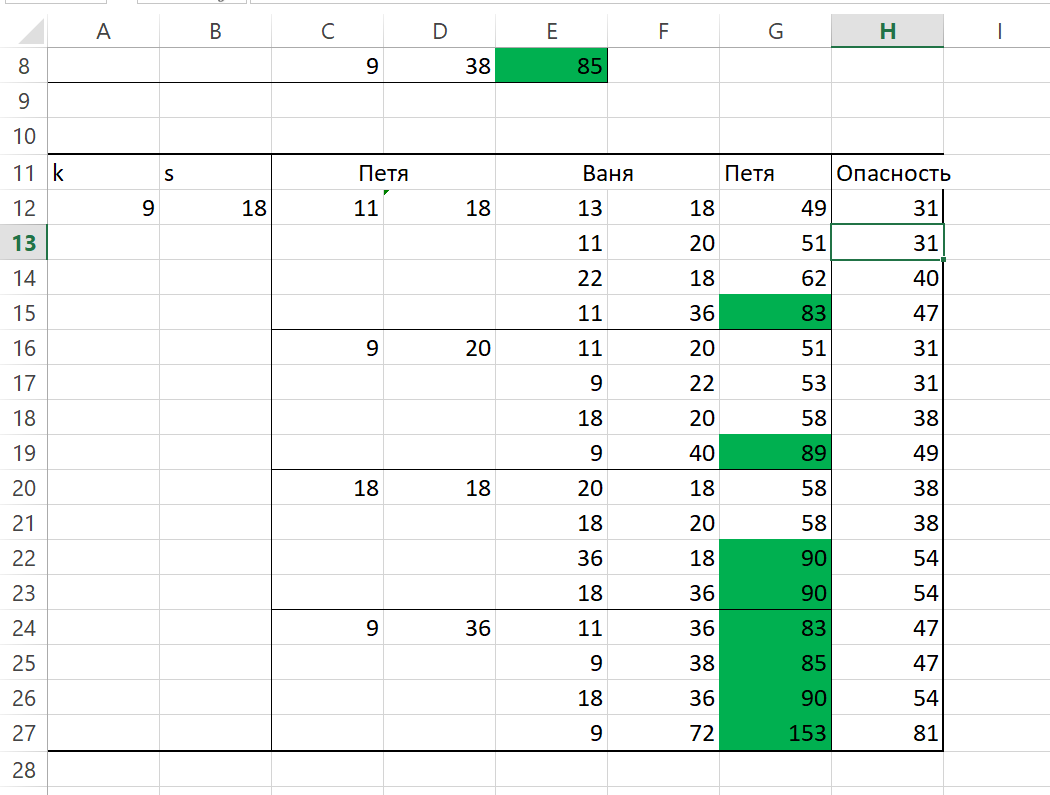
1. Дальше мы должны подставлять разные значения S (То есть перебрать значения). И смотреть исход игры.
   1. Чтобы упросить себе жизнь и не смотреть на выполнение условий, мы сделаем следующие.



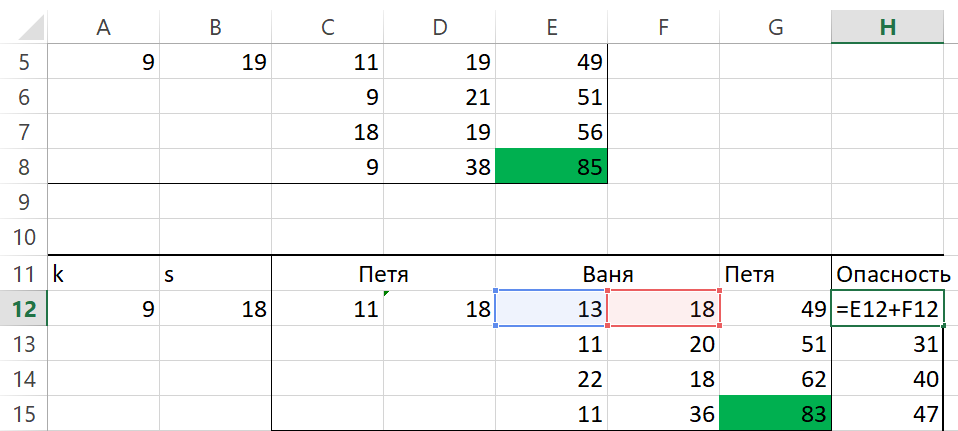
Ответ: 45

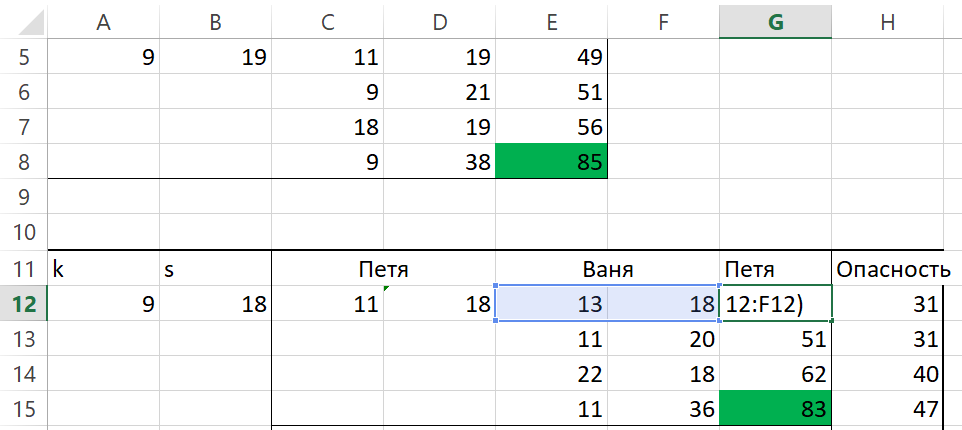
20. **Ходы: +2; \*2**

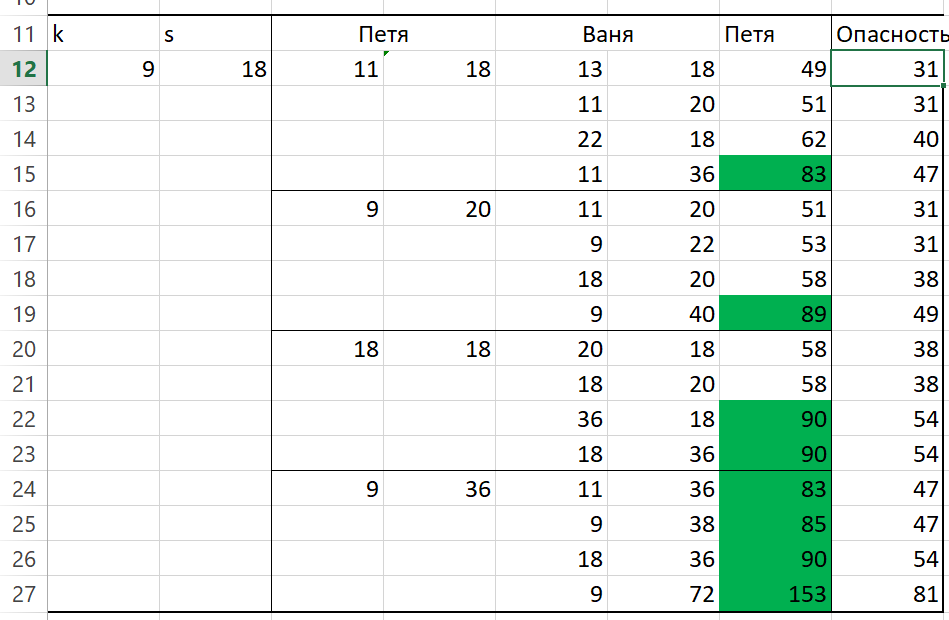
**Игра закончиться при S> =82 камней.**

**Тут мы должны уже рассматривать больше вариантов, а также искать случаи, когда игрок может проиграть.**

1. По аналогии обозначаем условие и заполняем ячейки с кучами
2. Мы должны рассмотреть на каждый ход Пети **ВСЕ ХОДЫ** Вани, а также обозначим новую ячейку с именем “Опасность”. Там мы будем складывать ходы Ваня, для выявление проигрышной позиции. Формула в G12 “=МАКС(E12:F12) \* 2+ МИН(E12:F12)”, В H12 = “=E12+F12”;





Что должно получиться.

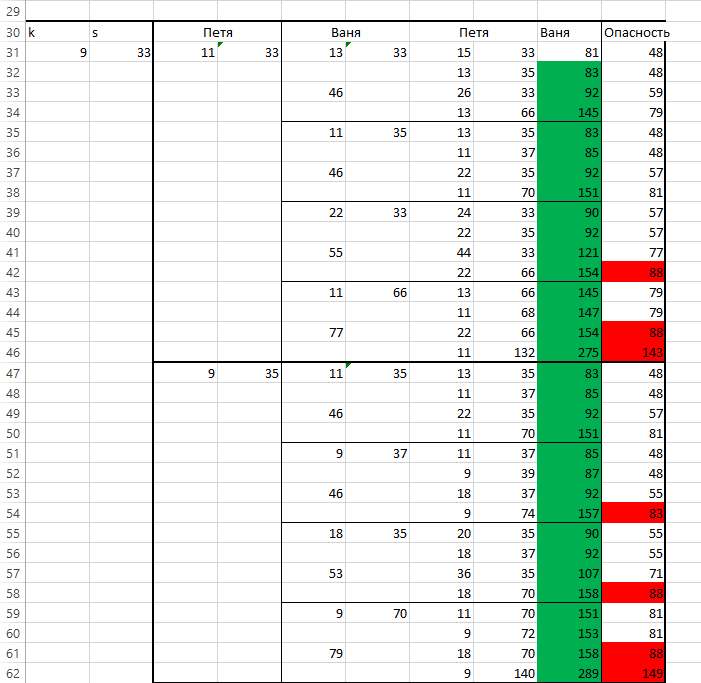
Мы также должны **перебирать S.** И смотреть, что все условию будут выполнены.

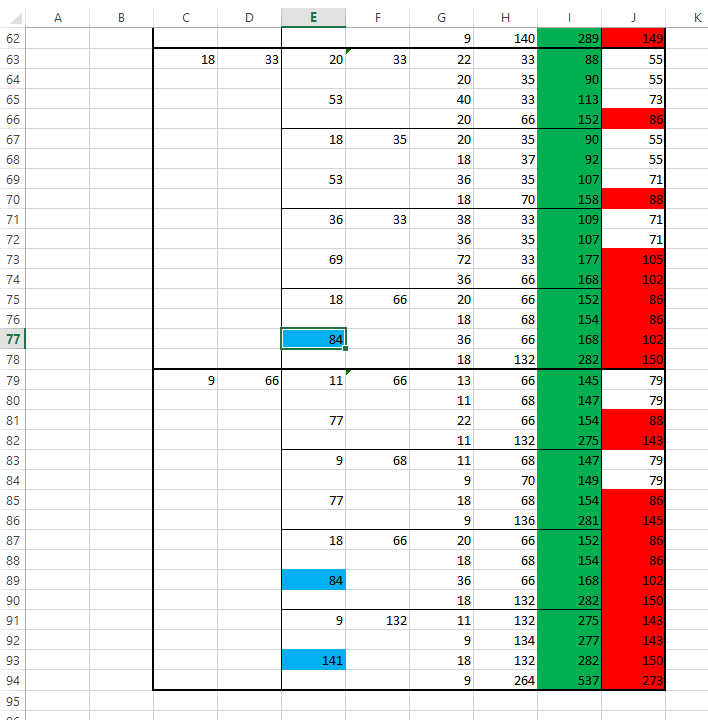
Ответ: 46 90

21. **Ходы: +2; \*2**

**Игра закончиться при S> =82 камней.**

1. **Скопируем прошлую таблицу и переместим заголовки!**
2. **Впишем все возможные ходы, и на каждый ход мы должны предусмотреть всевозможные ходы Пети. И должны не забыть про выигрышную стратегию Вани.**

****



Вы, наверное, обратили внимание, что некоторые ячейки голубого цвета, это неспроста! Обозначили мы их для того, чтобы сразу выявить победу первым ходом. (Значение должно быть >32)

1. Следующий шаг будет перебор значений S. Опять же нужно смотреть, удовлетворяет ли все значения условию 21 задания.

Ответ: 48

Python

ЗАДАЧА (ПРИМЕР) САЙТ ПОЛЯКОВА(https://kpolyakov.spb.ru)

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 один камень** 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 69. Победителем считается игрок, сделавший последний ход, т. е. первым получивший позицию, в которой в кучах будет 69 или больше камней.  
В начальный момент в первой куче было 5 камней, во второй куче – S камней, 1 ≤ S ≤ 63. 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. Найдите два таких 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 3. Укажите минимальное 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.

**Перед этим разбором мы настоятельно рекомендуем ознакомиться с рекурсией.**

**pythontutor.ru/lessons/functions**

**Вопрос 1**

Функция f принимает 3 аргумента: x - число камней в первой куче, y - число камней во второй куче, h - чей ход (1 - начало игры, 2 - ход Пети, 3 - ход Вани, 4 - ход Пети и т.д.)

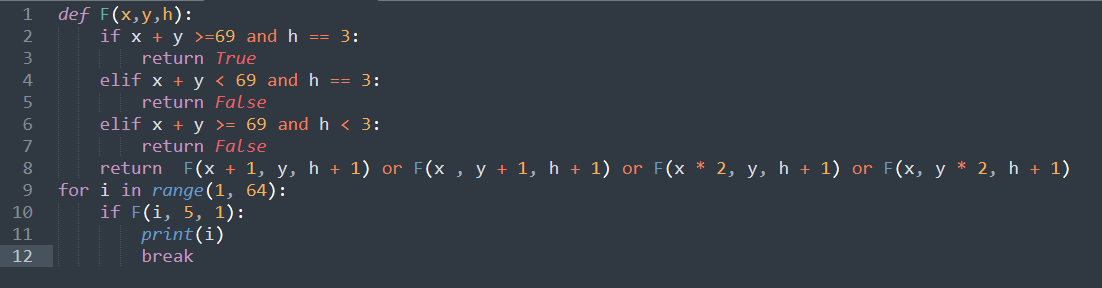
Строки 2-3. Если число камней в двух кучах больше или равно 69 и ход Вани, то вернуть истину.

Строки 4-5. Если ход Вани, но число камней в куче меньше 69, то вернуть ложь.

Строки 6-7. Если выиграл кто-то раньше, чем ход Вани, то вернуть ложь.

Строка 8. Продолжаем игру: функция вызывает саму себя по четырем направлениям, увеличивая при этом ход. Функция OR служит связкой этих направлений и означает "Хотя бы одно из этих направлений должно выдать истину". Если бы в условии было сказано, что "Ваня должен выиграть своим первым ходом при любом ходе Пети", то мы бы использовали функцию AND вместо OR.

Строки 10-12. Подбираем число X (число камней в первой куче), если при таком значении функция выдает истину, печатаем его. (Промежуток, который указан в коде, мы взяли не из головы, а из задачи)

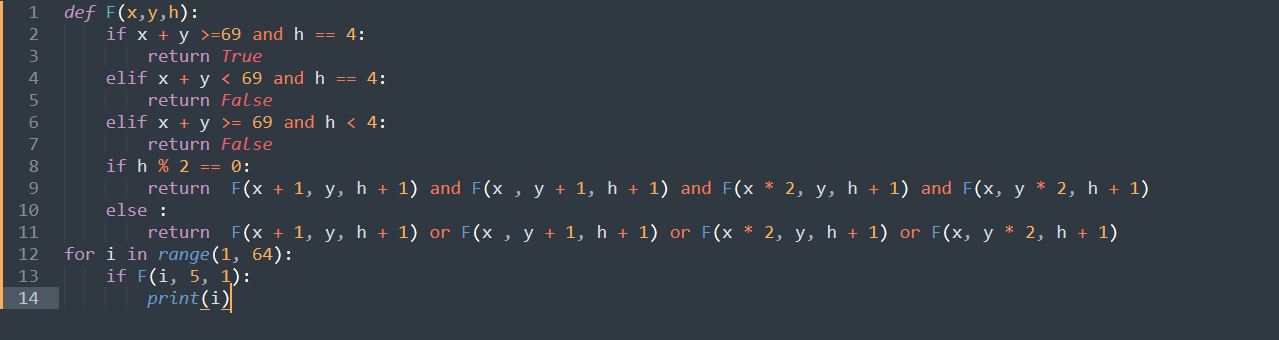


**Вопрос 2**

Для ответа на второй вопрос нужно немного изменить нашу программу. Петя должен выиграть вторым ходом (h == 4)! Следующие перечитываем вопрос!

*“…два значения S… Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня…”*

Значит, что Петя ходит как ему нужно (достаточно одного правильного хода из четырёх), а Ваня - как ему вздумается (все четыре хода должны привести Петю к выигрышу). Ходы Пети чётные (2, 4), ходы Вани - нечётные (3, 5). Значит после нечётного h (после Вани ходит Петя) мы будем использовать "или" (OR), а после чётного - "и" (AND).



**Вопрос 3**

Теперь мы должен выиграть Ваня первым или вторым ходом (h == 3 или h == 5). Значит, Ваня ходит так, как ему нужно, а Петя - как ему вздумается. Ваня ходит после чётного (

1-Исходное; 2-Петя; 3-Ваня…, таким образом, если h чётный, то пишем "или", а если нечётный - "и".

