

Понятие «выигрышная стратегия».

Разбор задач ЕГЭ тип 19,20,21.

УЧИТЕЛЬ ИНФОРМАТИКИ  
МАОУ «ЦО №1» Г. БЕЛГОРОДА  
ВОРОНИНА ЕКАТЕРИНА ЕВГЕНЬЕВНА

## Введение. Теория игр

- заключается в анализе стратегического взаимодействия двух и более сторон (игроков), а также последствий принятия этими сторонами решений. Под игрой понимается любой процесс, в котором несколько участников идут к цели и имеют личный интерес. Это может быть компьютерная стрелялка, переговоры о повышении зарплаты, предвыборная гонка или даже выбор удобного времени для совещания.

### Суть теории игр.

**Основная цель теории игр** — понять, как игроки могут оптимизировать свои стратегии, чтобы достичь наилучших результатов.

*Некоторые принципы:*

Рациональность игроков — каждый игрок стремится максимизировать свою выгоду, принимая решения на основе доступной информации.

Стратегическое взаимодействие — результат действий одного игрока зависит от действий других игроков.

Моделирование конфликтов и сотрудничества — теория игр позволяет анализировать как конкурентные, так и кооперативные ситуации.

## Методы

Задания 19–21 ЕГЭ по информатике посвящены теории игр с кучами камней. Для решения этих задач можно использовать разные методы, например:

Анализ алгоритма игры. Нужно понять, как работает игра и каковы последовательности ходов игроков.

Поиск выигрышной стратегии. Необходимо определить, какой ход гарантирует победу игроку, независимо от действий соперника.

Построение дерева игры и определение стратегии победы. Нужно рассмотреть все возможные варианты развития игры и выбрать оптимальную стратегию.

Некоторые способы решения:

- ✓ Использование кода на Python. С его помощью можно быстро анализировать все возможные игровые позиции и находить правильные ответы.
- ✓ Применение электронных таблиц. Можно использовать, например, Excel, OpenOffice, LibreOffice.
- ✓ Аналитический или графический способ.

## Рассмотрим задачу 19. Условия игры:

Два игрока Петя и Ваня играют в игру

□Игроки ходят по очереди, начинает Петя. За ход можно:

1.Убрать 5 камней.

2.Если камней чётное количество, уменьшить их в 2 раза.

3.Если камней кратно 3, уменьшить их в 3 раза.

4.Если камней нечётное и не кратно 3, добавить 1 камень.

Пример: если в куче 12 камней, то за один ход можно получить 7,6,4 камня, а если 11 камней то за один ход получаем 6 или 12 камней

□Игра заканчивается, когда камней  $S < 19$  □Побеждает тот, кто сделал последний ход.

Цель: Найти минимальное  $S > 19$ , при котором:

□Петя не может выиграть первым ходом (т.е. все его возможные ходы ведут к позиции, где Ваня может выиграть первым ходом).

□Ваня может выиграть первым ходом из любой позиции, достигнутой Петей.

## Код программы

```
def f(s,m):  
    if s<19: return m%2==0  
    if m==0: return 0  
    h=[f(s-5,m-1)]  
    if s%2==0: h=[f(s//2, m-1)]  
    if s%3==0: h=[f(s//3, m-1)]  
    if s%2!=0 and s%3!=0: h=[f(s+1,m-1)]  
    return any(h) if (m-1)%2!=0 else any(h)  
  
print ('19',[s for s in range(19,1000) if not f(s,1) and  f(s,2)])
```

## Задание 20.

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

```
def f(s,m):  
    if s<19: return m%2==0  
    if m==0: return 0  
    h=[f(s-5,m-1)]  
    if s%2==0: h=[f(s//2, m-1)]  
    if s%3==0: h=[f(s//3, m-1)]  
    if s%2!=0 and s%3!=0: h=[f(s+1,m-1)]  
    return any(h) if (m-1)%2!=0 else any(h)  
print ('20',[s for s in range(20,1000) if not f(s,1) and f(s,3)])
```

## Задание 21.

Для игры, описанной в задании 19, найдите минимальное 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.

```
def f(s,m):  
    if s<19: return m%2==0  
    if m==0: return 0  
    h=[f(s-5,m-1)]  
    if s%2==0: h=[f(s//2, m-1)]  
    if s%3==0: h=[f(s//3, m-1)]  
    if s%2!=0 and s%3!=0: h=[f(s+1,m-1)]  
    return any(h) if (m-1)%2!=0 else any(h)  
print ('21',[s for s in range(20,1000) if f(s,2) and f(s,4)])
```

Спасибо за внимание!