

# Игровая задача ЕГЭ

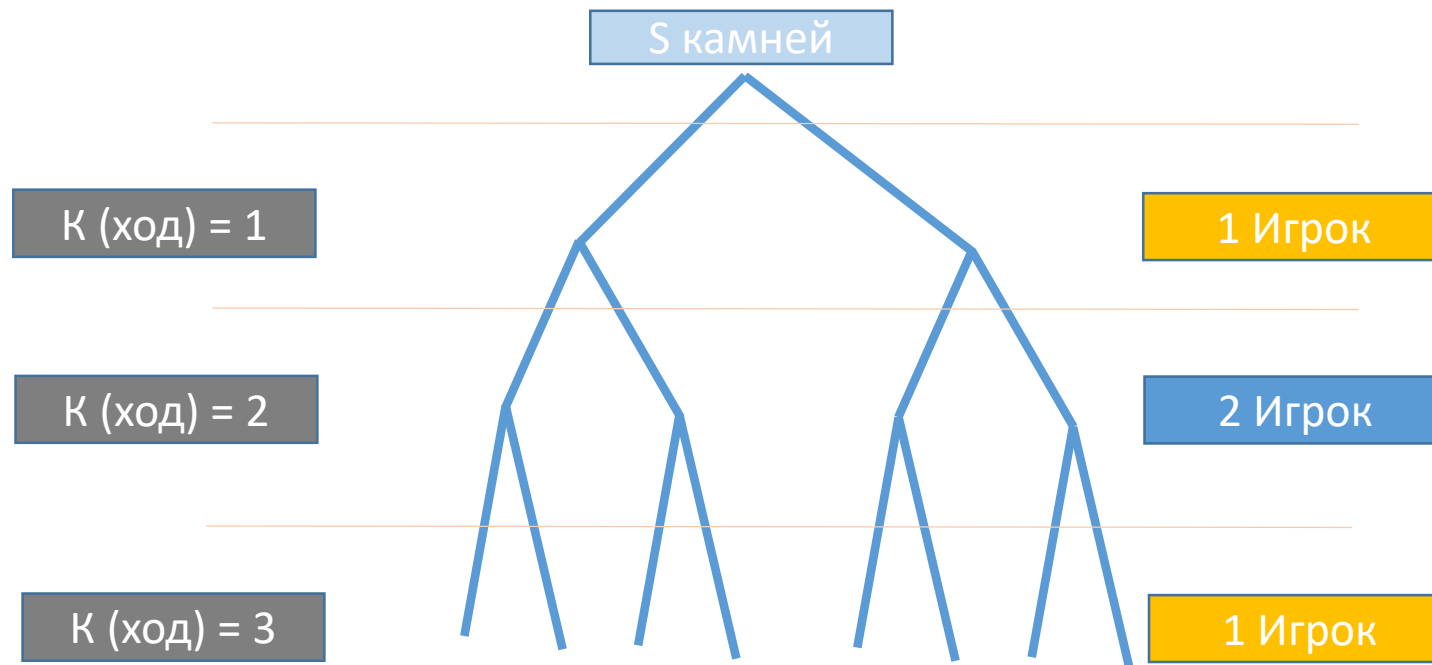
Решение с помощью кода методом построения дерева игры и оценки узлов

# Правила игры (наиболее стандартные на ЕГЭ)

Игроки ходят по очереди, увеличивая количество камней в куче.  
(количество камней у каждого игрока бесконечно).

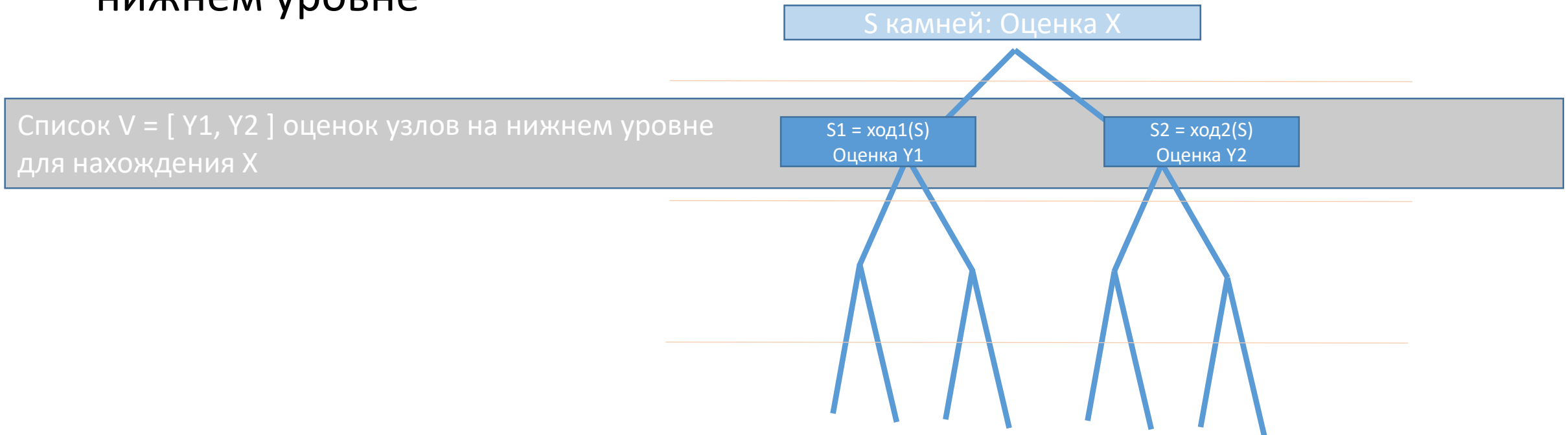
Побеждает тот, кто первым набирает необходимую сумму.

Дерево игры (с двумя вариантами ходов игрока):



# Правила оценки узлов дерева

1) Оценить узел можно **только при известных оценках** узлов на нижнем уровне

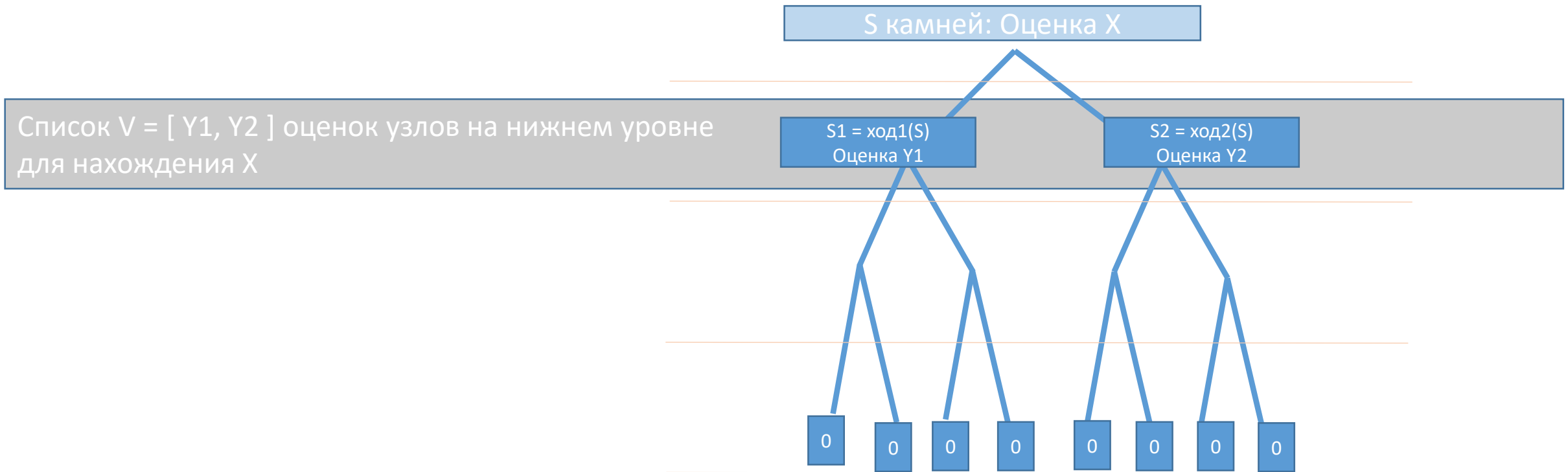


- Таким образом, алгоритм будет рекурсивным (обход графа)

# Правила оценки узлов дерева

2) Оценка узла равна 0

Количество камней в куче стало достаточным для победы (игрок, сделавший такое значение, победил)

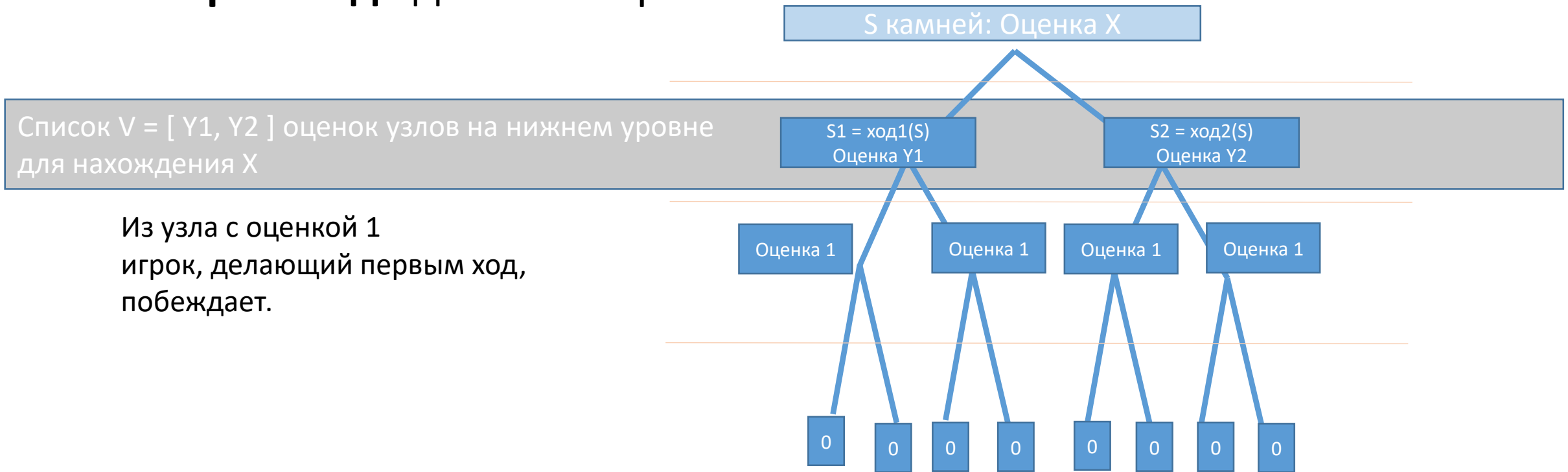


- Фактически оценка 0 – оценка листа дерева – условие завершения рекурсивного спуска.

# Правила оценки узлов дерева

3) Оценка узла может быть положительной или отрицательной:

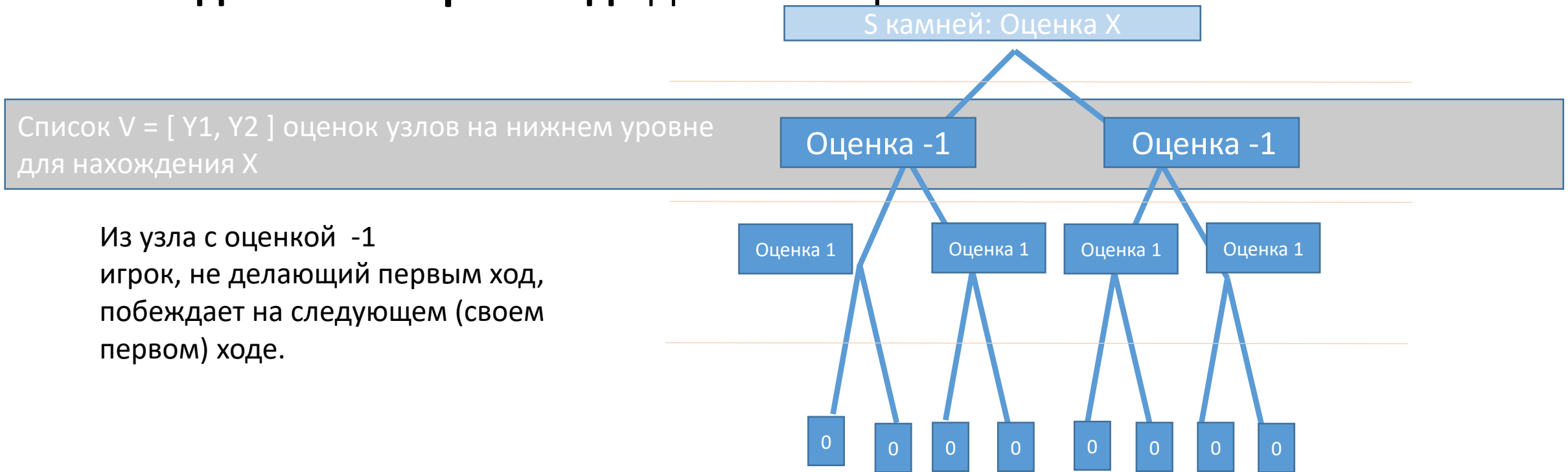
А) Если оценка **положительна** – выигрывает игрок, **начинающий** играть из данного узла. Значение оценки совпадает с **победным номером хода** данного игрока



# Правила оценки узлов дерева

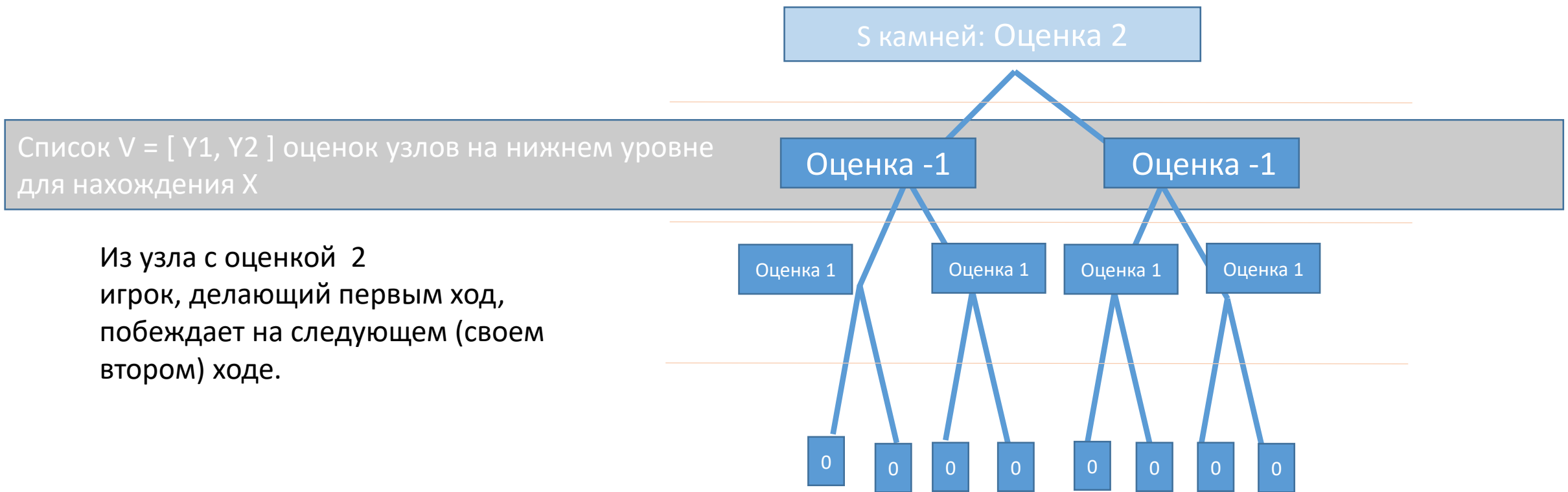
3) Оценка узла может быть положительной или отрицательной:

Б) Если оценка **отрицательная** – выигрывает игрок, **не начинающий** играть из данного узла. Значение оценки совпадает с **победным номером хода** данного игрока



# Правила оценки узлов дерева

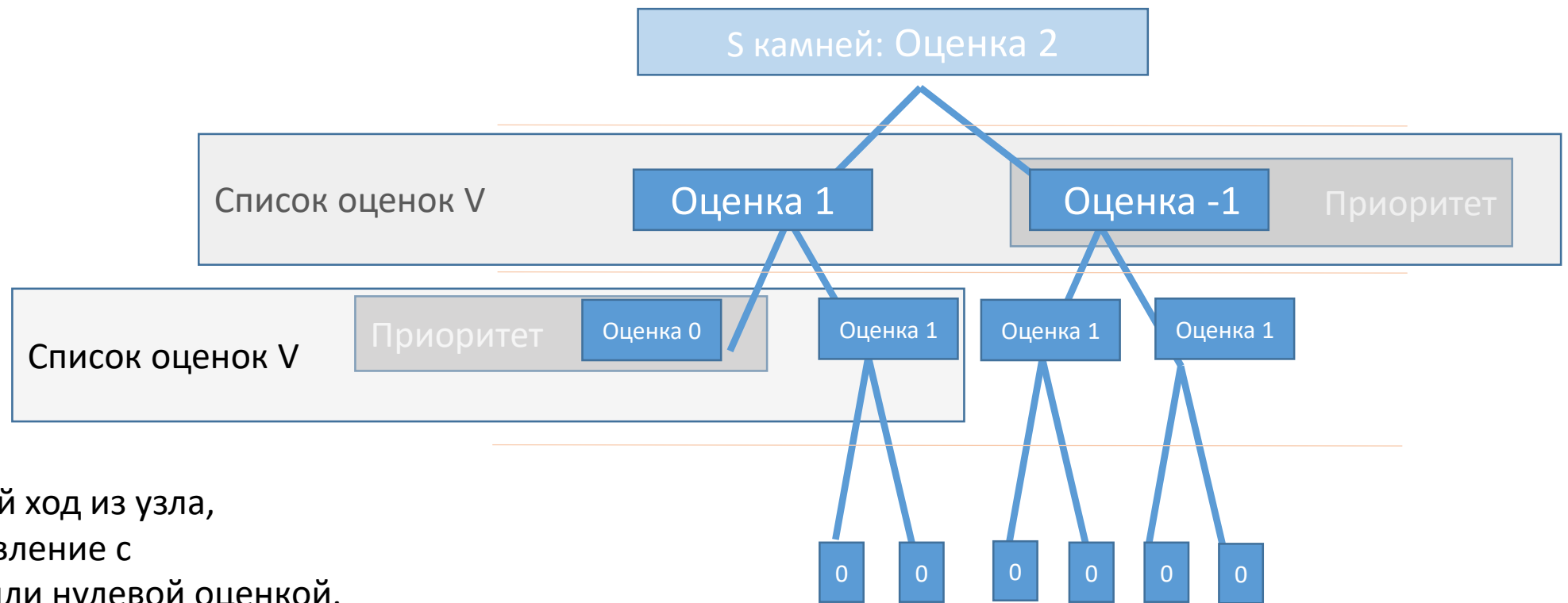
3) Оценка узла может быть положительной или отрицательной.



# Правила оценки узлов дерева

4) Стратегия заключается в поиске приоритетных направлений движения по дереву, исходя из полученных оценок

Приоритет отдается нулевым или отрицательным оценкам.



Игрок, делающий ход из узла, выбирает направление с отрицательной или нулевой оценкой.

# Правила оценки узлов дерева

## 5) Формулы нахождения оценки узла из оценок узлов на нижнем уровне.

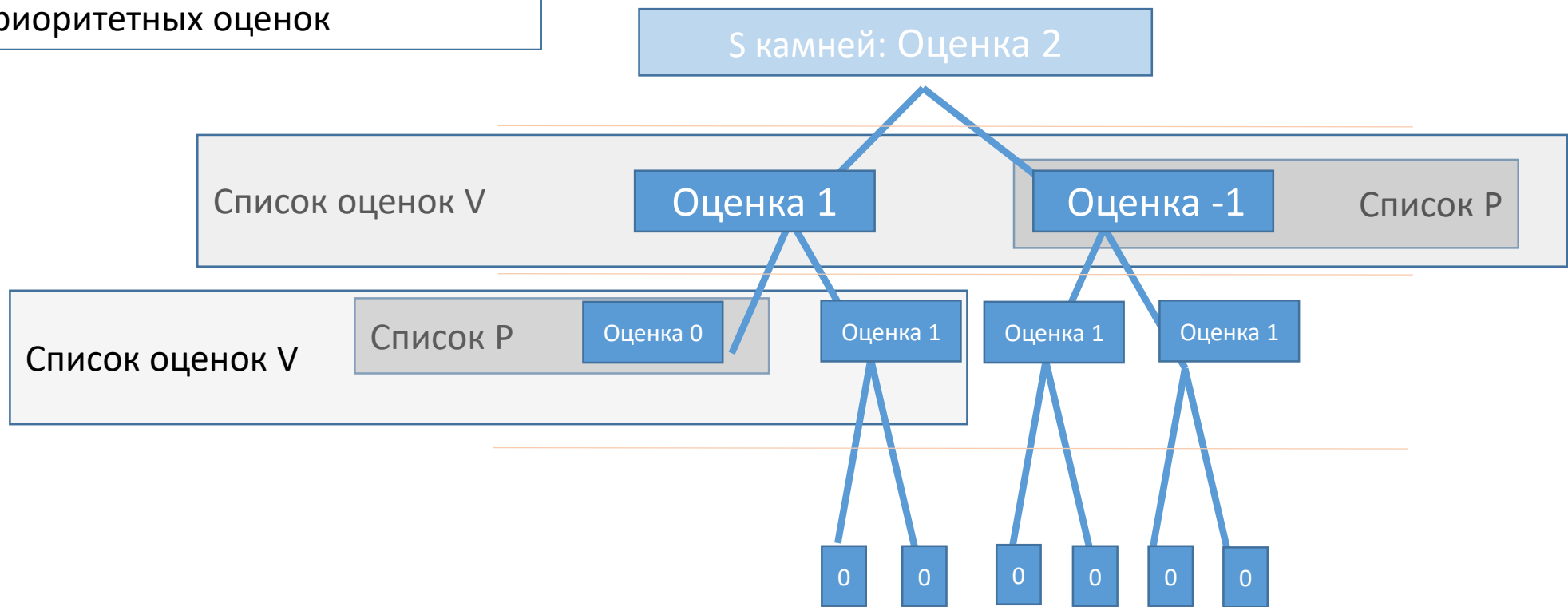
$X$  – Оценка узла

$V$  – Список всех оценок на нижнем уровне

$P$  – Список приоритетных оценок

Если  $P$  не пуст, то оценка  $X = -\max(P) + 1$

Если  $P$  пуст, то оценка  $X = -\max(V)$

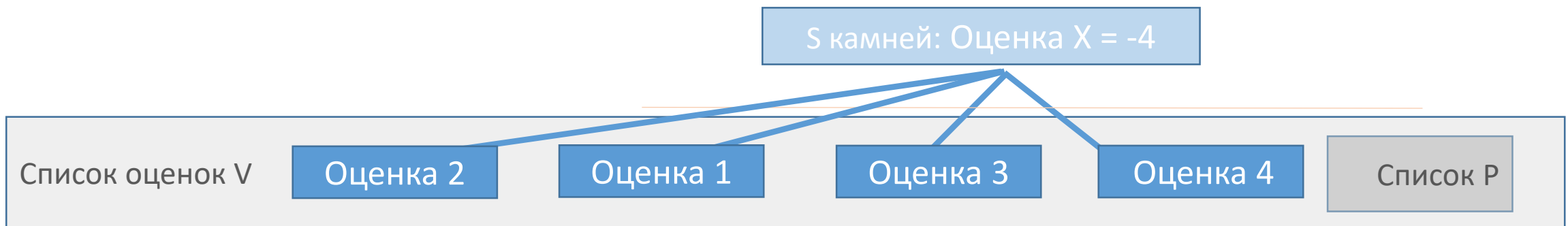
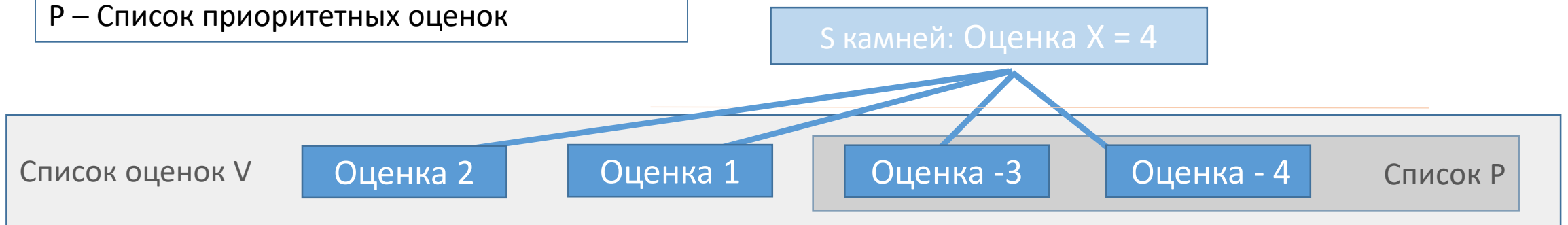


# Правила оценки узлов дерева

## 5) Формулы нахождения оценки узла из оценок узлов на нижнем уровне.

$X$  – Оценка узла  
 $V$  – Список всех оценок на нижнем уровне  
 $P$  – Список приоритетных оценок

Если  $P$  не пуст, то оценка  $X = -\max(P) + 1$   
Если  $P$  пуст, то оценка  $X = -\max(V)$



# Код на примере задачи из демоверсии 2026

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 3 камня;
- убрать из кучи 5 камней;
- уменьшить количество камней в куче в 4 раза (количество камней, полученное при делении, округляется до меньшего).

Например, из кучи в 20 камней за один ход можно получить кучу из 17, 15 или 5 камней.

Игра завершается, когда количество камней в куче становится не более 30. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу из 30 или менее камней. В начальный момент в куче было  $S$  камней,  $S \geq 31$ .

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

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

## Задание 20.

Для игры, описанной в задании 19, найдите два наименьших значения  $S$ , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

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

Найденные значения запишите в ответе в порядке возрастания.

## Задание 21.

Для игры, описанной в задании 19, 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.

## Результаты для каждой задачи

```
# 19 задача
for s in range(31, 1000):
    r = game(s)
    if r == -1:
        print(s)

# 20 задача
for s in range(31, 1000):
    r = game(s)
    if r == 2:
        print(s)

# 21 задача
for s in range(31, 1000):
    r = game(s)
    if r == -2:
        print(s)
```

Функция, возвращающая ходы игрока из узла (количество камней из некоторой точки  $X$ )

```
def move(x):
    return [x - 3, x - 5, x // 4]
```

Рекурсивная функция, строящая дерево с вычислением оценок узлов. Главная задача – нахождение оценки стартового узла

```
def game(x, k = 0):  2 usages
    if k > 4 or x <= 30:
        return 0
    v = [game(y, k + 1) for y in move(x)]
    p = [y for y in v if y <= 0]
    if p:
        return -max(p) + 1
    return -max(v)
```

$S$  камней: Оценка  $X = -4$

Список оценок  $V$

Оценка 2

Оценка 1

Оценка 3

Оценка 4

Список  $P$