

## Invariant solutions

---

1. [Blackboard game - [http://problems.ru/view\\_problem\\_details\\_new.php?id=60630](http://problems.ru/view_problem_details_new.php?id=60630)]

**Invariant value:** the result of multiplication of all numbers on the blackboard does not change with any valid move.

Since initial result of multiplication was  $-1$ , the last number on the blackboard will be  $-1$

2. [Coin game [http://problems.ru/view\\_problem\\_details\\_new.php?id=30767](http://problems.ru/view_problem_details_new.php?id=30767)]

Since the machine takes one coin and gives out 5, the number of coins Luize has changes in increments of 4.

**Invariant value** in this case is number of coins modulo 4 - throughout the process, amount of counts divided by 4 always gives the same remainder.

Since initial amount is congruent 1 modulo 4, but required end state 2018 is congruent 2 modulo 4, then it is clearly unreachable.

3. [Knight chess game - [http://problems.ru/view\\_problem\\_details\\_new.php?id=102984](http://problems.ru/view_problem_details_new.php?id=102984)]

No it is not.

Notice, that a knight always moves from white to black and from black to white fields.

**Invariant value** - after each odd numbered move, knight will be on a white field, and after each even-numbered move, it will be on the black field.

There are 64 fields on a chess board, but since the knight is already on one of the fields, it should make 63 moves to have visited all of them once. 63 is an odd number, so it will land on a white field, but  $h8$  is a black field, so it is impossible.

4. [Stones game]

In each move we are decreasing the total amount of stones on the table by 1 and increasing the number of stone piles by 1. Therefore...

**Invariant value** is a sum of stones and piles. Then the contradiction is easy to see, for example, as  $3 \times n + n = 4n$  stones at the end state means that starting state should be  $0 \bmod 4$

5. [Circle game - [http://problems.ru/view\\_problem\\_details\\_new.php?id=88309](http://problems.ru/view_problem_details_new.php?id=88309)]

If you color the numbers in two colors -  $A$  and  $B$ , the coloring is uninterrupted, meaning that any two neighboring numbers will contain one of each color.

**Invariant value** in this case is un/equality of sums of numbers in each color.

Since we are always adding 1 to two numbers, one of each color, if we start out with unequal sums, then they will stay unequal no matter what moves we make. However the required end position has equal sums, so it is unreachable.

6. [PAMO2018PL6]

**Answer:** all  $n = 3k + 1$  and  $n = 3k + 2$

**Proof that it works for  $n = 3k + 1$  and  $n = 3k + 2$**

An algorithm how to do it in following steps:

a. One move takes us from  $(1, 1, \dots, 1, 0, 1, 1, \dots)$  to  $(1, 1, \dots, 0, 1, 0, 1, \dots, 1)$

b. One move allow us to increase string of 0's like this:  $(\underbrace{0, \dots, 0, 0}_m, \overbrace{1, 0, 1, 1, 1, \dots}^{\text{move}}) \rightarrow (\underbrace{0, \dots, 0, 0}_{m+1}, \overbrace{1, 0, 1, 1, \dots})$

c. Using two first rules (first - once, second -  $3k$  times), we can get to  $(0, \dots, 0, 1, 0, \dots, 0)$ . For  $n = 3k + 1$  goto last step.

## Invariant solutions

---

- d. In case of  $n = 3k+2$  we apply move once like this  $(0, \dots, 0, 0, \overbrace{1, 0, 0}^{\text{move}}, 0, 0, \dots) \rightarrow (0, \dots, 0, 0, \overbrace{0, 1, 1}, 0, 0, \dots)$
- e. At this moment we have left with a cluster of one or two 1's and a string of  $3k$  zeros, which we can convert to all 1 with  $k$  moves:  $(\dots, \overbrace{0, 0, 0}, \dots) \rightarrow (\dots, \overbrace{1, 1, 1}, \dots)$

### **Proof that it can not be done for $3k$**

We color the numbers in three colors -  $A$ ,  $B$  and  $C$  cyclically. Since  $n = 3k$  coloring cycle is uninterrupted.

Notice that any move swaps exactly one number of each color.

**Invariant value:** Equality of parity of sums of numbers in each color. I.e. if the parity sum of all  $A$  numbers was the same/opposite as those of color  $B$ , it will stay the same/opposite during any valid move(s).

Now simply note that in beginning position parity of one color is different from other two, but required end configuration has equal parities in all colors, so it is impossible to reach it.