Games and Processes - Solutions

1. *[IMO2009SLC1]*

Problem

Consider 2009 cards, each having one gold side and one black side, lying in a line on a long table. Initially all cards show their gold sides. Two players, standing by the same long side of the table, play a game with alternating moves. Each move consists of choosing a block of 50 consecutive cards, the leftmost of which is showing gold, and turning them all over, so those with showed gold now show black and vice versa. The last player who can make a legal move wins.

a. Does the game necessarily end?

b. Does there exist a winning strategy for the starting player?

Solution - binary representation; strong variant; coloring a subset; analysing whole ws inductuion

a. If we denote gold cards by 1, and black by 0, the entire sequence of cards corresponds to a number in binary representation. After each of the moves, the number decreases, hence the game has to end.

b. We will show that second player wins a game no matter how the players play. Consider the cards whose position (counted from the right) is divisible by 50. There is a total of 40 such cards, and in each move exactly one of this cards is turned over. In the beginning, all 40 of these cards are 1, and in the end all 40 are 0, hence the second player must win.

2. *[IMO2015SLC1]*

Problem

Several positive integers are written in a row. Iteratively, Alice chooses two adjacent numbers x and y such that $x > y$ and x is to the left of y, and replaces the pair (x, y) by either $(y + 1, x)$ or $(x{-}1, x)$.

Prove that she can perform only finitely many such iterations.

Solution - bounded variant; digit representation

Firstly observe that maximum element M of this sequence does not change during allowed operations.

Now, consider a *characteristic* C of particular sequence v of integers in our problem. Value of C would be calculated by inverting string v and then interpreting elements of this invcerted sequence as digits in M-base counting system.

I.e. if original v consists of k elements $v_1, v_2, \ldots, v_k$ then $C = v_k \times M^{k-1} + v_{k-1} \times M^{k-2} + v_{k-2} \times M^{k-3} + \cdots + v_2 \times M^1 + v_1 \times M^0$

It is easy to see that every allowed operation strictly increases C, but on other hand C is upper bound by $M^k - 1$, so the process must stop at some point.

3. $[IMO2011SLC3]$

Problem

Let S be a finite set of at least two points in the plane. Assume that no three points of S are collinear. By a windmill we mean a process as follows. Start with a line l going through a point $P \in S$. Rotate l clockwise around the pivot P until the line contains another point Q of S. The point Q now takes over as the new pivot. This process continues indefinitely, with the pivot always being a point from S. Show that for a suitable $P \in S$ and a suitable starting line l containing P, the resulting windmill will visit each point of S as a pivot infinitely often.

Solution - continuous process, geometry, invariant

Give the rotating line an orientation and distinguish its sides as the *oranje* side and the *blue* side. Notice that whenever the pivot changes from some point T to another point U, after the change, T is on the same side as U was before. Therefore, the number of elements of S on the oranje side and the number of those on the blue side remain the same throughout the whole process (except for those moments when the line contains two points).

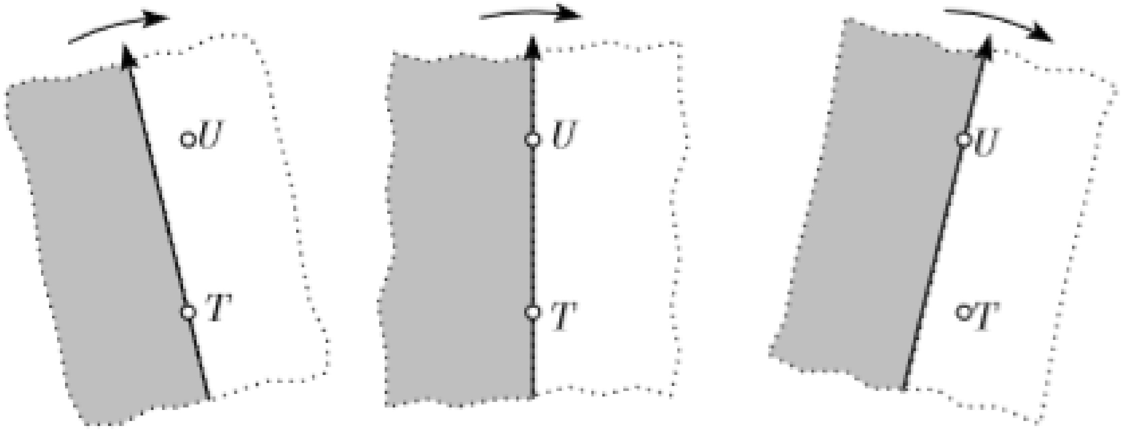

Figure 1: Number of points staying constant on each side of the windmill

First consider the case that $|S| = 2n + 1$ is odd. We claim that through any point $T \in S$, there is a line that has n points on each side. To see this, choose an oriented line through T containing no other point of S and suppose that it has $n + r$ points on its oranje side. If $r = 0$ then we have established the claim, so we may assume that $r \neq 0$. As the line rotates through 180° around T, the number of points of S on its oranje side changes by 1 whenever the line passes through a point; after 180°, the number of points on the oranje side is $n—r$. Therefore there is an intermediate stage at which the oranje side, and thus also the blue side, contains n points.

Now select the point P arbitrarily, and choose a line through P that has n points of S on each side to be the initial state of the windmill. We will show that during a rotation over 180°, the line of the windmill visits each point of S as a pivot. To see this, select any point T of S and select a line l through T that separates S into equal halves. The point T is the unique point of S through which a line in this direction can separate the points of S into equal halves (parallel translation would disturb the balance). Therefore, when the windmill line is parallel to l, it must be l itself, and so pass through T.

Next suppose that $|S| = 2n$. Similarly to the odd case, for every $T \in S$ there is an oriented line through T with $n{-}1$ points on its oranje side and n points on its blue side. Select such an oriented line through an arbitrary P to be the initial state of the windmill.

We will now show that during a rotation over 360°, the line of the windmill visits each point of S as a pivot. To see this, select any point T of S and an oriented line l through T that separates S into two subsets with $n{-}1$ points on its oranje and n points on its blue side. Again, parallel translation would change the numbers of points on the two sides, so when the windmill line is parallel to l with the same orientation, the windmill line must pass through T.

Comment One may shorten this solution in the following way.

Suppose that $|S| = 2n + 1$. Consider any line l that separates S into equal halves; this line is unique given its direction and contains some point $T \in S$. Consider the windmill starting from this line. When the line has made a rotation of 180°, it returns to the same location but the oranje side becomes blue and vice versa. So, for each point there should have been a moment when it appeared as pivot, as this is the only way for a point to pass from on side to the other.

Now suppose that $|S| = 2n$. Consider a line having $n{-}1$ and n points on the two sides; it contains some point T. Consider the windmill starting from this line. After having made a rotation of 180°, the windmill line contains some different point R, and each point different from T and R has changed the color of its side. So, the windmill should have passed through all the points.

4. $[IMO2009SLC5]$

Problem

Five identical empty buckets of 2-liter capacity stand at the vertices of a regular pentagon. Cinderella and her wicked Stepmother go through a sequence of rounds: At the beginning of every round, the Stepmother takes one liter of water from the nearby river and distributes it arbitrarily over the five buckets. Then Cinderella chooses a pair of neighboring buckets, empties them into the river, and puts them back. Then the next round begins. The Stepmother's goal is to make one of these buckets overflow. Cinderella's goal is to prevent this.

Can the wicked Stepmother enforce a bucket overflow?

*What should be the volume of buckets for answer to change?

Solution - complex invariant

No, the Stepmother cannot enforce a bucket overflow and Cinderella can keep playing forever. Throughout we denote the five buckets by B_0, B_1, B_2, B_3, and B_4, where B_k is adjacent to bucket B_{k-1} and B_{k+1} $(k = 0, 1, 2, 3, 4)$ and all indices are taken modulo 5. Cinderella enforces that the following three conditions are satisfied at the beginning of every round:

a. Two adjacent buckets (say B_1 and B_2) are empty.

b. The two buckets standing next to these adjacent buckets (here B_0 and B_3) have total contents at most 1.

c. The remaining bucket (here B_4) has contents at most 1.

These conditions clearly hold at the beginning of the first round, when all buckets are empty.

Assume that Cinderella manages to maintain them until the beginning of the r-th round $(r \geq 1)$. Denote by $x_k (k = 0, 1, 2, 3, 4)$ the contents of bucket B_k at the beginning of this round and by y_k the corresponding contents after the Stepmother has distributed her liter of water in this round.

By the conditions, we can assume $x_1 = x_2 = 0, x_0 + x_3 \leq 1$ and $x_4 \leq 1$. Then, since the Stepmother adds one liter, we conclude $y_0 + y_1 + y_2 + y_3 \leq 2$. This inequality implies $y_0 + y_2 \leq 1$ or $y_1 + y_3 \leq 1$. For reasons of symmetry, we only consider the second case.

Then Cinderella empties buckets B_0 and B_4. At the beginning of the next round B_0 and B_4 are empty (condition (a) is fulfilled), due to $y_1 + y_3 \leq 1$ condition (b) is fulfilled and finally since $x_2 = 0$ we also must have $y_2 \leq 1$ (condition (c) is fulfilled).

Therefore, Cinderella can indeed manage to maintain the three conditions (a)–(c) also at the beginning of the $(r+1)$-th round. By induction, she thus manages to maintain them at the beginning of every round. In particular she manages to keep the contents of every single bucket at most 1 liter. Therefore, the buckets of 2-liter capacity will never overflow.

5. $[IMO2010SLC4]$

Problem

Six stacks $S_1, \ldots, S_6$ of coins are standing in a row. In the beginning every stack contains a single coin. There are two types of allowed moves:

a. If stack S_k with $1 \leq k \leq 5$ contains at least one coin, you may remove one coin from S_k and add two coins to S_{k+1}.

b. If stack S_k with $1 \leq k \leq 4$ contains at least one coin, then you may remove one coin from S_k and exchange stacks S_{k+1} and S_{k+2}.

Decide whether it is possible to achieve by a sequence of such moves that the first five stacks are empty, whereas the sixth stack S_6 contains exactly $2010^{2010^{2010}}$ coins.

Solution - inductuion

Denote by $(a_1, a_2, \ldots, a_n) \to (a'_1, a'_2, \ldots, a'_n)$ the following: if some consecutive stacks contain $a_1, \ldots, a_n$ coins, then it is possible to perform several allowed moves such that the stacks contain $a'_1, a'_2, \ldots, a'_n$ coins respectively, whereas the contents of the other stacks remain unchanged.

Let $A = 2010^{2010^{2010}}$. Our goal is to show that

$$(1,1,1,1,1,1) \to (0,0,0,0,0,A)$$

First we prove two auxiliary observations.

Lemma 5.1. $(a, 0, 0) \to (0, 2^a, 0)$ for every $a \geq 1$.

Proof. We prove by induction that $(a, 0, 0) \to (a - k, 2k, 0)$ for every $1 \leq k \leq a$.

For $k = 1$, apply Move a to the first stack:

$$(a, 0, 0) \to (a - 1, 2, 0) = (a - 1, 2^1, 0)$$

.

Now assume that $k < a$ and the statement holds for some $k < a$. Starting from $(a - k, 2^k, 0)$, apply Move a to the middle stack $2k$ times, until it becomes empty. Then apply Move b to the first stack:

$$(a - k, 2^k, 0) \to (a - k, 2k - 1, 2) \to \cdots \to (a - k, 0, 2^{k+1}) \to (a - k - 1, 2^{k+1}, 0)$$

Hence,

$$(a, 0, 0) \to (a - k, 2^k, 0) \to (a - k - 1, 2^{k+1}, 0)$$

■

Lemma 5.2. For every positive integer n, let $P_n = \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{n}$ (e.g.$P_3 = 2^{2^2} = 16$)

Then $(a, 0, 0, 0) \to (0, P_a, 0, 0)$ for every $a \geq 1$.

Proof. Similarly to Lemma 5.1, we prove that $(a, 0, 0, 0) \to (a - k, P_k, 0, 0)$ for every $1 \leq k \leq a$. For $k - 1$, apply Move 1 to the first stack:

$$(a, 0, 0, 0) \to (a - 1, 2, 0, 0) = (a - 1, P_1, 0, 0)$$

Now assume that the lemma holds for some $k < a$. Starting from $(a - k, P_k, 0, 0)$, apply Lemma 5.1, then apply Move b to the first stack:

$$(a - k, P_k, 0, 0) \to (a - k, 0, 2^{P_k}, 0) = (a - k, 0, P_{k+1}, 0) \to (a - k - 1, P_{k+1}, 0, 0)$$

Therefore,

$$(a, 0, 0, 0) \to (a - k, P_k, 0, 0) \to (a - k - 1, P_{k+1}, 0, 0)$$

■

Now we prove the statement of the problem.

First apply Move a to stack 5, then apply Move b to stacks S_4, S_3, S_2 and S_1 in this order. Then apply Lemma 5.2 twice:

$$\begin{aligned}(1,1,1,1,1,1) &\to (1,1,1,1,0,3) \to (1,1,1,0,3,0) \to (1,1,0,3,0,0) \to (1,0,3,0,0,0) \\ &\to (0,3,0,0,0,0) \to (0,0,P_3,0,0,0) = (0,0,16,0,0,0) \to (0,0,0,P_{16},0,0)\end{aligned} \tag{1}$$

We already have more than A coins in stack S_4, since

$$A \leq 2010^{2010^{2010}} \leq (2^{11})^{2010^{2010}} = 2^{11 \times 2010^{2010}} < 2^{2010^{2011}} < 2^{(2^{11})^{2011}} = 2^{2^{11 \times 2011}} < 2^{2^{2^{15}}} < P_{16}$$

To decrease the number of coins in stack S_4, apply Move b to this stack repeatedly until its size decreases to $A/4$. (In every step, we remove a coin from S_4 and exchange the empty stacks S_5 and S_6.)

$$(0,0,0,P_{16},0,0) \to (0,0,0,P_{16}-1,0,0) \to (0,0,0,P_{16}-2,0,0) \to \cdots \to (0,0,0,A/4,0,0)$$

Finally, apply Move a repeatedly to empty stacks S_4 and S_5:

$$(0,0,0,A/4,0,0) \to \cdots \to (0,0,0,0,A/2,0) \to \cdots \to (0,0,0,0,0,A)$$

6. $[IMO2012P3]$

Problem

The liar's guessing game is a game played between two players A and B. The rules of the game depend on two positive integers k and n which are known to both players.

At the start of the game the player A chooses integers x and N with $1 \leq x \leq N$. Player A keeps x secret, and truthfully tells N to the player B. The player B now tries to obtain information about x by asking player A questions as follows: each question consists of B specifying an arbitrary set S of positive integers (possibly one specified in some previous question), and asking A whether x belongs to S. Player B may ask as many questions as he wishes. After each question, player A must immediately answer it with yes or no, but is allowed to lie as many times as she wants; the only restriction is that, among any $k+1$ consecutive answers, at least one answer must be truthful.

After B has asked as many questions as he wants, he must specify a set X of at most n positive integers. If $x \in X$, then B wins; otherwise, he loses. Prove that:

(a) If $n \geq 2^k$ then B has a winning strategy.

(b) There exists a positive integer k_0 such that for every $k \geq k_0$ there exists an integer $n \geq 1.99^k$ for which B cannot guarantee a victory.

Solution - interesting binary search, ... TODO - needs improvement

Solution to Part (a)

It suffices to show that there is a winning strategy for $n = 2^k - 1$, as a winning strategy for any $n \geq 2^k$ easily gives a winning strategy for $n = 2^k - 1$.

Player B first divides all integers 1 through N into 2^k nonempty sets T_s, where s is some binary string of length k. On the i^{th} question for $1 \leq i \leq k$, player B asks whether the set containing x is labeled with a string that has a 1 in the i^{th} position. After asking these questions, there is one set that player A must indicate does not contain x at k times in a row; WLOG we may assume this set is $T_{00\ldots0}$.

On the next question, B asks if $x \in T_{00\ldots0}$. If A says 'no', then B can rule out $T_{00\ldots0}$. If A says 'yes', then B asks if $x \in T_{100\ldots0}$. If A says 'no', then B can rule out $T_{100\ldots0}$. If A says 'yes', then B asks if $x \in T_{0100\ldots0}$. If A says 'no', then B can rule out $T_{0100\ldots0}$, while if A says 'yes', B can rule out $T_{1100\ldots0}$.

In any case, B can ascertain that $x \notin T_s$ for some string s, thus reducing the number of possibilities for x. Player B can repeat this procedure as long as at least 2^k elements remain, eventually reducing the number of possible values of x to at most $2^k - 1$.

Solution to Part (b)

Take k sufficiently large so that there is some positive integer n with $n \geq 1.99^k$ but $n + 1 \leq .001 * 1.999^{k+1}$.

Player A sets $N = n + 1$. Let $f(i, m)$ denote the number of times in a row that A has just indicated that $i \neq x$. In other words, if A indicates on the m^{th} step that x was in a set containing i, then $f(i, m) = 0$, while if A indicates otherwise, we would have $f(i, m) = f(i, m-1) + 1$.

At the m^{th} step, player A makes a decision so as to minimize the sum:

$$g(m) = \sum_{i=1}^{N} 1.999^{f(i,m)}$$

I claim that, by using this strategy, A will preserve the invariant that $g(m) < 1.999^{k+1}$. We can prove this by induction on m. Note that initially, $g(0) = N < 1.999^{k+1}$.

Suppose that $g(m-1) < 1.999^{k+1}$. Note that if B asks if $x \in S$ for some $S \subset \{1, 2, \ldots, N\}$, then the two possibilities for $g(m)$ are:

$$g_{\text{in}}(m) = N - |S| + 1.999 \sum_{i \in S} 1.999^{f(i,m-1)}$$

and

$$g_{\text{out}}(m) = |S| + 1.999 \sum_{i \notin S} 1.999^{f(i,m-1)}$$

Note that $g_{\text{in}}(m) + g_{\text{out}}(m) = N + 1.999g(m-1)$. Therefore, we have that:

$$\min(g_{\text{in}}(m), g_{\text{out}}(m)) \leq \frac{g_{\text{in}}(m) + g_{\text{out}}(m)}{2} = \frac{N + 1.999g(m-1)}{2} < \frac{.001 * 1.999^{k+1} + 1.999^{k+2}}{2} = 1.999^{k+1}$$

This completes the induction. In particular, since $g(m) < 1.999^{k+1}$ at all times, A will never indicate that $x \neq i$ more that k times in a row. So player B will never be able to rule out any element of $\{1, 2, \ldots, N\}$ and submit a set of n integers that are guaranteed to contain x.