Olympiads in Informatics, 2008, Vol. 2, 115–121

Naturalness in Tasks for Olympiads in Informatics

Pavel S. PANKOV
International University of Kyrgyzstan
A. Sydykov str., 252, Apt. 10, 720001 Bishkek, Kyrgyzstan
e-mail: pps50ster@gmail.com, pps50@rambler.ru

Abstract. There are two main ways to invent tasks for olympiads of a high level: one way is to invent or choose an effective algorithm and compose a corresponding subject, and another way is to think out a real situation or task and try to formalize it. The second way is more difficult because it is multi-stage: the author needs to find some effective algorithm for a task obtained; if the best algorithm is obvious or the only algorithm seems to be exponential then we need to rework the formulation, etc. But by our opinion the second way is preferable because it can yield original tasks with natural, short and elegant formulations and give less advantage to experienced participants. We shall consider the second way in detail in this paper.

Key words: olympiads in informatics, tasks, naturalness.

1. Natural Ways to Generate Tasks

Diks (2007) mentioned the following phases of the task preparation process: review of task ideas, formulation, analysis, verification and calibration. This sequence may be applied not only to informatics but to other sciences as well. Since we have not found publications on generating of task ideas in informatics, we shall review a way to generate ideas, basing on actions in spaces which we call *natural,* and include our own experience.

1.1. *Choosing a Space*

Firstly, a space is to be chosen. Traditionally, there are used the following types of spaces:

S1) integer numbers (one-dimensional grid);
S2) pairs of integer numbers (plane grid) (often, it is introduced as ”grid of streets in the host town of the olympiad”);
S3) triples of integer numbers (space grid), but tasks on such space are usually too difficult for solving;
S4) a graph;
S5) space of solids rolling on a plane (an intermediate between S2 and S3);
S6) ring (a segment with glued ends) with finite number of elements (an intermediate between S1 and S4).
We propose also:
S7) two connected rings (figure-of-eight);

Certainly, S7 is a kind of S4 but the general graph demands a vast description since ”figure-of-eight” is self-explanatory.

S8) integer grids on non-Euclidean spaces, e.g., topological torus, Moebius band (Weeks, 1985). They are defined not as manifolds in Euclidean space but as arrays with gluing of edges as follow:
Define a rectangular grid
$G = \{(X, Y) | 0 \leqslant X \leqslant M, 0 \leqslant Y \leqslant N; M, N, X, Y$ are integers$\}$.
A *Moebius band* is obtained from G when, for all $Y = 0, 1, \ldots, N$, points $(0, Y)$ and $(M, N - Y)$ are glued. A *topological torus* is obtained from G when, for all $Y = 0, 1, \ldots, N$, points $(0, Y)$ and (M, Y) are glued and, for all $X = 0, 1, \ldots, M$, points $(X, 0)$ and (X, N) are glued.

Since the 1970s, the topological torus has appeared in computer games naturally: if an object disappears beyond one of the edges of the screen then it appears from the opposite edge.

1.2. *Choosing Actors*

Further, ”actors” (moving, changing objects) are to be chosen. Involving more than one actor provides a *game*. But many interesting tasks with more than one actor could be generated without the idea of being games, for instance, by means of cooperation of actors. Something more, the organization connected with the proposal of a game-task during an olympiad in informatics (with preparing auxiliary libraries, describing interfaces, etc.) is too difficult. But some games can be imitated by means of formulating the aim of a task as minimax.

Traditionally, we have the following configurations of actors:

A1) a point is used as an actor (a point can move by 1 or ”jump” but the rules of jumping must be very simple);

A2) some rectangles.

We propose also:

A3) two or three points (or many points with very simple conditions);

A4) moving ”train” of a length of one edge or one arc within S4;

A5) moving ”train” of a given length within S2, S6, S7.

1.3. *Choosing Actions*

The main action proposed is natural *moving*. Moving a train along a graph or a grid is a consecutive passing of its vertices by the head of the train, by its intermediate points (if its length is greater than 1) and by its tail. Simple *cutting*, *gluing*, *deleting* and *adding* are also natural actions.

1.4. *Choosing Conditions, Restrictions and Obstacles*

Conditions may be natural, i.e., actors must/cannot

C1) coincide;
C2) pass;
C3) overlap;
C4) touch;
C5) be seen;
C6) cross;
C7) cross itself (for a train).
For A3 natural conditions are also:
C8) be near/far each from other.
Another natural kind of restrictions is the following:
C9) an actor can make only a given number of steps.
Traditional type of obstacles is a labyrinth but some points or rectangles can replace it.

1.5. *Aims and Composing of Tasks*

The aims may be as following:

G1) to reach/build/compose something in a minimal number of steps;
G2) to build/compose the least/greatest object;
G3) to find the shortest/longest way;
G4) to catch another actor; to escape from another actor in a minimal time; to escape with minimal number of steps or expenditures for all possible actions of another actor (i.e., if its behavior is optimal).

By describing the space, the actors with their possible actions, the conditions, restrictions and obstacles, and declaring the aim, we obtain a task of optimizing or determining if the goal is "impossible or possible" and optimizing if the goal is possible.

Even if the author is sure the aim is attainable in all cases of the task, the output meaning "impossible" must be provided in the description of all permitted outputs of the task.

2. Grading System

Certainly, the author must try to find the best algorithm to solve the task. If s/he is sure that the best algorithm has been found (constructed), s/he is to think about possible weaker algorithms or even wrong algorithms such as a "greedy" ones, and compose the set of tests (as it is described in some papers, Diks (2007)). Also, the author must keep in mind that sometimes the best algorithm cannot be found within a restricted time. For example, the task in the paper Pankov (2005) has a solution of time complexity $O(1)$ operations but it is impossible to find it during a few hours.

At the same time, if the author cannot find the best algorithm then the natural tasks give the following ways of composing the set of tests.

2.1. *"Evident" Solutions*

For some of natural tasks the answer (the best answer) is "seen" by a human for all initial data within given restrictions. If the author is certain that the answer is self-evident then s/he may compose as many sufficiently different cases as necessary covering all aspects of the task. A good solution (algorithm) by a competitor must solve many of them. In other words, we propose the hypothesis: if the answers for all initial data are evident for a human then there exists a good algorithm solving the task on the modern computer during an appropriate time.

2.2. *Open-Ended Tasks*

Grading of such tasks is considered, for example (Kemkes, 2007). We will mention some known procedures.

Compose as many as necessary sufficiently different cases covering all aspects of the task. Consider them for a task on maximization. Let M be the maximal number of points for any test.

One of the ways is to write a simple algorithm yielding any boundaries A_-, A_+ for the (unknown) result. If $A \leqslant A_-$, then a result A obtains 0 points else $(M(A-A_-)/(A_+ - A_-)$ rounded down) points.

Another way is to compare the result with the records of all other contestants. Let $A_{\max}$ be the best result of all contestants in this test. If $A \leqslant A_{\max}/2$ then a result A obtains 0 points else ($MA/A_{\max}$ rounded down) points.

3. Examples of Tasks

Some of tasks built by means of above techniques were published in (Pankov, 2000; 2003; 2007). Tasks 1, 2, 3, 4 listed below are generalizations of ones given at olympiads in informatics of different levels in Kyrgyzstan in 2004–2008.

Task 1. Given a graph, its vertices are "houses" (less than 7). The Instrument has counted mice under each of houses at different moments. During all this measuring, each mouse could pass to another neighbor house only once. Write a program to find the least possible number of mice.

Example. Six houses form a ring. Input: 9, 0, 1, 0, 0, 2. Output: 10. [Two mice under the first house and two mice under the sixth one could be the same].

Generation: S4 or S5; A3; C9 (one step); G2.

Task 2. A graph is given. Firstly, the head H and the tail T of a train are in two neighbor vertices. Write a program finding one of the shortest ways to be passed by the train (moving forward only) in order to put its head to the primary position of T and its tail to the primary position of H.

Example. The graph contains vertices A, B, C, D and edges AB, BC, BD, CD. Firstly, $HT = BA$. One of the shortest ways for H is $B - C - D - B - A$.

Generation: S4; A4; G3.

Task 3. Let the streets in the city form a rectangular grid (of given size). The firm Logic [sponsor] is situated at a given crossing (X, Y). Two friends wish to come to the firm. Now the first is at the crossing $(X1, Y1)$, the second is at the crossing $(X2, Y2)$. Because of plentiful snowing they wish to minimize the trampled path (the sum of paths trampled by the first, by the second and by the both going together). Write a program calculating the minimal length of such path.

Generation: S4; A3; G3.

Task 4. At night, a mouse is anywhere within a long ditch of ”figure-of-eight” of length 2008 meters, the first ring of the ditch is numbered from 0 till 1004 (from the cross to the cross) and the second ring is numbered from 1004 till 2008 (the points with numbers 0, 1004 and 2008 coincide). The mouse can run quickly but cannot climb out. Two men with sacks stand at given points $X1$ and $X2$. The men's velocity is 1 meter/second. Write a program calculating the minimal time to catch the mouse in any case.

Example. Input: $X1 = 500,\ X2 = 504$; output: 1006.

Generation: S7; A3; C5; G4.

Task 5. A piece of the upper half-plane is cut by broken (not self-touching) line connecting N points: $(X[1], Y[1] = 0), (X[2], Y[2] > 0), \ldots, (X[N-1], Y[N-1] > 0), (X[N], Y[N] = 0)$. Given $(2N-2)$ integer numbers $X[1], X[2], Y[2], \ldots, X[N-1], Y[N-1], X[N]$, write an algorithm detecting whether this piece can be extracted from the half-plane by means of motion within the plane.

Generation idea: what surfaces can be punched? But “surfaces” are too difficult; after discussion they were changed to “figures” in S2.

Comment: The author had in mind but did not write ”by means of parallel shift” because he was sure that the motion to extract could be parallel shift only and including these words would be a prompting.

This task was given at the III All-USSR olympiad in Informatics, held in Kharkov at 1990, and was solved by pen-and-paper. During the olympiad, all organizers and contestants also thought that the only possible motion was a parallel shift and all algorithms investigated possibility of such a motion only. But just after the closing ceremony one of the contestants found an example of ”upper half of a (narrow) crescent” which can be extracted by rotation!

This task demonstrates both dangers and interest arising while implementing the proposed approach.

4. Proposals and Conclusion

We propose some tasks built within the proposed approach.

Task 6. Given a graph (with less than 10 vertices) and two sets B(initial positions of wanderers) and E of its vertices, $|E| \geqslant |B|$. Write a program finding the least number of steps to move all wanderers from B to E under the condition that they do not meet each other.

Generation: S4; A3; C8; G3.

Comment: This is one of the simplest examples of possibilities of A3.

Task 7. Consider a rectangular grid $0 \leqslant X \leqslant M, 0 \leqslant Y \leqslant N$.

A) For all $Y = 0, 1, \dots, N$, points $(0, Y)$ and $(M, N - Y)$ are the same; or

B) For all $Y = 0, 1, ..., N$, points $(0, Y)$ and (M, Y) are the same; for all $X = 0, 1, \dots, M$, points $(X, 0)$ and (X, N) are the same.

Write a program calculating

C) the shortest way between two given points $(X1, Y1)$ and $(X2, Y2)$; or

D) the shortest cycle connecting three given points $(X1, Y1),\ (X2, Y2)$ and $(X3, Y3)$ along the grid.

Generation: S8; G3.

Comment: This is an example of possibilities of S8.

Task 8. The head H of a train of length N is at the point $(0, N)$ and its tail T is at the point $(0, 0)$. The train can move (forward only) along edges of the rectangular grid (pairs of integer numbers) not self-touching and cannot pass given points $(X1, Y1), (X2, Y2), \dots, (XK, YK)$. Write a program finding one of the shortest ways to be passed by the train in order to put H at the point (XH, YH) and at the same time to put T at the point (XT, YT). Main restriction: $|XH - XT| + |YH - YT| \leqslant N$.

Generation: S2; A5; C4; G3.

Task 9. Cut a given rectangle with integer sides by two segments parallel to its sides (to three or four rectangles with integer sides) and

A) compose of these rectangles (without overlapping) a polygon (with all angles right) of the least possible perimeter; or

B) shift and overlap them to compose a polygon of the least possible area. (Only parallel shift is permitted).

Generation: S2; A2; C3; G2.

Let us demonstrate composing a task on a given theme. There is the Fig. 1 *Regions that hosted finals of the NOI (15 towns)* in the paper Dagiene (2007). *Idea*: “Two friends with bicycles decided to make photos of these towns for the illustrated history of the NOIs”. Choose the endpoints: “Now (in the morning) they are in Vilnius (the capital; 16th town) and must return here”. Further, the array of distances (may be, in hours rather than kilometers) between some pairs of these 16 points must be given. Also, choose the time necessary for making photos in every town (for instance, 3 hours). To make the task more realistic, add: “one can ride or make photos not more 12 hours a day.” Thus, we obtain the *Task 10*: “Write a program calculating the minimal number of days for such enterprise under given conditions”.

Generation: S4; A3; C2; G3.

We hope that tasks built in such a way would yield short and elegant formulation (Dagiene, 2007), would be interesting for young people and attractive for prospective sponsors. Also, such tasks give less advantage to experienced participants because they would not be able to use known algorithms immediately. Analysis of programs written by contestants within conditions of Subsection 2.1 as it was proposed by Verhoeff (2006) and is seen from *Task 5* would yield interesting and unexpected results.

References

Dagiene, V., Skupiene, J. (2007). Contests in programming: quarter century of Lithuanian experience. *Olympiads in Informatics* , **1**, 37–49.

Diks, K., Kubica, M., Stencel, K. (2007). Polish olympiads in informatics: 14 years of experience. *Olympiads in Informatics* , **1**, 50–56.

Kemkes, G., Cormack, G., Munro, I., Vasiga, T. (2007). New task types at the Canadian computing competition. *Olympiads in Informatics* , **1**, 79–89.

Pankov, P., Acedanski, S., Pawlewicz, J. (2005). Polish flag. In: *The 17th International Olympiad in Informatics (IOI'2005). Tasks and Solutions*. Nowy Sacz, 19–23.

Pankov, P.S., Oruskulov, T.R. (2007). Tasks at Kyrgyzstani olympiads in informatics: experience and proposals. *Olympiads in Informatics* , **1**, 131–140.

Pankov, P.S., Oruskulov, T.R., Miroshnichenko, G.G. (2000). *School Olympiads in Informatics (1985–2000 years)*. Bishkek (in Kyrgyz & Russian).

Pankov, P.S., Oruskulov, T.R., Miroshnichenko, G.G. (2003). *Olympiad Tasks in Informatics, Devoted to Kirghiz Statehood, History of Kyrgyzstan and Great Silk Road*. Bishkek (also in Kyrgyz & Russian).

Verhoeff, T. (2006). The IOI is (not) a science olympiad. *Informatics in Education*, **5**(1), 147–159.

Weeks, J.R. (1985). *The Shape of Space*. Marcel Dekker, Inc., New York.

P.S. Pankov (1950), doctor of physical-math. sciences, prof., corr. member of Kyrgyzstani National Academy of Sciences (KR NAS), is the chairman of Jury of Bishkek City OIs since 1985, of Republican OIs since 1987, the leader of Kyrgyzstani teams at IOIs since 2002. Graduated from the Kyrgyz State University in 1969, is a main research worker of Institute of Mathematics of KR NAS, a manager of chair of the International University of Kyrgyzstan.