Olympiads in Informatics, 2022, Vol. 16, 125–133

DOI: 10.15388/ioi.2022.10

Olympiads in Informatics in Kyrgyzstan

Pavel S. PANKOV[1], Kylychbek A. URAIYMOV[2],
Artem A. BELYAEV[3]

[1]*Institute of Mathematics, Kyrgyzstan*
[2]*International Educational Institution "Sapat", Kyrgyzstan*
[3]*Kyrgyz-Russian Slavic University, Kyrgyzstan*
e-mail: pps5050@mail.ru, kylychbek.uraimov@sapat.edu.kg, artem_belyaev@mail.ru

Abstract. This report provides description of the Multi-stage National Olympiad along with other inner competitions in informatics, the online platform to conduct competitions in informatics in Kyrgyzstan, the participation in IOI and in other international olympiads in informatics.

This report also presents the principles and ways to create tasks (naturalness; presentation of real processes; reference to known objects; non-Euclidean spaces; "brute force method is either inapplicable or gives too overestimate of complexity") and examples of tasks of various levels with classification.

Keywords: Kyrgyzstan, National Olympiad, informatics, task, site.

1. Introduction

The conduct of Olympiads in informatics in Kyrgyzstan since 1985 was described (Pankov *et al.*, 2007; Pankov *et al.*, 2011).

- **Section 2** contains the description of the multi-stage National Olympiad, other competitions in informatics in Kyrgyzstan, participation in IOI and other international olympiads in informatics.
- **Section 3** describes the online platform to conduct competitions in informatics in Kyrgyzstan.
- **Section 4** presents a survey of principles and ways to create tasks (naturalness; presentation of real processes; reference to known objects; non-Euclidean spaces; "brute force method is either inapplicable or gives too overestimate of complexity").
- **Section 5** contains examples of tasks of various levels created with methods of Section 4.

2. Competitions

Teaching informatics (under the traditional name "Foundations of Informatics and Computer Facilities") in secondary schools of Kyrgyzstan started in 1985.

The Olympiads in Bishkek city, the capital of Kyrgyzstan, are conducted since 1985 (annually in January).

Republican Olympiads in informatics, as a constituent of Republican Olympiads in various subjects of secondary schools are conducted since 1987.

Stage I is schools' (November); stage II is rayons' (December); stage III is: 7 regions, Bishkek city and Osh city (February or March); stage IV is final (March).

(Now Republican Olympiad in informatics is a constituent of National Olympiad).

Selection competitions to IOI for prize-winners of IV stage (April) were conducted in English since 2000 until 2019.

National Olympiads in informatics, as a constituent of selection to International Olympiads (IOI, IMO, IPhO, IChO, IBO) are conducted since 2020.

Stage I is for all comers, online (January); stage II is online (February); stage III is on-site (March); stage IV (selection to IOI) is for prize-winners of stage III and of the final stage of Republican Olympiad, on-site (April).

Kyrgyzstan participates in IOI since 2000. Our achievements are bronze medals at IOI'2000, IOI'2004, IOI'2005, IOI'2016, IOI'2017, three bronze medals at IOI'2019, four bronze medals at IOI'2021.

Every year we conduct ICPC Kyrgyzstan Championship for students as a quarter-final of International Collegiate Programming Contest (November). Schoolchildren's teams participate in it out of competition.

Since 2015 our teams participate in International Zhautykov Olympiads (IZhO), our achievements in "Computer science" nomination are: 2015: 3 bronze; 2016: silver; 2017: gold, silver, bronze; 2018: silver, 2 bronze; 2019: 3 bronze; 2020: 2 bronze; 2021: 5 bronze; 2022: 2 gold, 4 bronze.

Since 2015 our teams participate in Asia-Pacific Informatics Olympiads (APIO), our achievements are: 2017: bronze; 2018: 3 bronze; 2019: silver; 2020: bronze; 2021: 2 bronze.

European Junior Olympiad in Informatics (EJOI), 2021: gold.

XIII Eurasian Olympiad in Informatics, Central Asia, 2021: first place.

3. The Online Platform

All competitions in informatics are conducted on the base of the site:

`olymp.krsu.edu.kg`

The site supports providing competitions of various types: according to the ICPC rules or student competitions; when a complete solution of the task is required or a partial solution is possible.

The system provides the following features:

Possibility to judge the solutions on different physical computers; automatic registration of the competitors; separation of access privilege to the contests; support of any programming languages (C++, C#, Java, Python, Pascal etc.) and the ability to add new languages easy setup without changing the application code; automatic verification solutions for malicious code; automatic re-checking solutions by the administrator's signal; limited allocation of computing resources (CPU time, random-access store, external memory) for solutions; statistical analysis of the problems solving process; virtual competitions support.

Architecture Components: Dispatcher (distribution of solutions between judge services), Judges (checking solution; compilers management); Web application and database.

Hence, the system has two components:

1. The site or web application itself, through which the user interacts with the system.
2. Dispatcher and judges who check and evaluate solutions.

Since these components are independent, they can be scaled. Solution verification can be run on completely different machines, which provides flexibly manage_of the load during various competitions. The task format is compatible with the `polygon.codeforces.com` system. It provides preparing problems and using ready-made problems without additional modifications.

4. Principles and Ways to Develop Tasks

We try to prepare tasks to be well-understood, to have "short and elegant formulations" (Dagienė *et al.*, 2007).

The following items are not defined exactly; they are overlapping somewhere. Certainly, we do not pretend to originality. We use such papers as (Burton *et al.*, 2008), (Diks *et al.*, 2008), (Kemkes *et al.*, 2007) and

4.1. Naturalness (Pankov, 2008) includes: presentation of real processes (Pankov, 2010) with gravitation, repelling and attracting; a human can „see" the answer for corresponding task image with initial data of small volume without calculations; tasks are difficult to be solved even with initial data of small volume; a „brute force" method is either inapplicable or gives too overestimation of complexity. Sometimes it is not necessary to write a program for generation of tests for a task. A human can compose sufficiently complex tests where the answer is "seen" but it is difficult to find it by means of any program.

4.2. Reference to known objects (Pankov *et al.*, 2009) including local circumstances, the host town, the host state, sponsors (at the same time, the task should be "culturally neutral").

4.3. For solving an Olympiad task by the contestant, in addition to the common limits in the CPU time (traditionally 1 second) and in memory (traditionally 256 megabyte) there exists an actual limit (*) on average time to write and debug a program even if the contestant has necessary skills and vision of the algorithm (about 1–1.5 hours). Sometimes such limit is achieved by means of a long-winded and complicated text of the task, with many "permissions" and "bans". In the proposed tasks this limit is achieved naturally, because of their geometrical content.

4.4. Unusual spaces in tasks are built sometimes by null-transportation etc. We propose to use "natural" non-Euclidean spaces: Moebius band, Riemann surfaces, topological torus, projective plane.

4.5. Using "regular" graphs instead of "general" graphs involves very vast graphs with short and well-understood description, for instance (combination with 4.4): Take a 2022 × 2022 square grid and add arcs: (1,1)-(2022,2022), (1,2)-(2022,2021), … (1,2022)-(2022,1) (Moebius band).

4.6. More than one actor or non-point actor within a graph.

4.7. Pseudo-game. How many moves are necessary to defeat an actor operating by a known (declared in the text of the task) algorithm? Attempts to write such a program are more effective than a "logical" way to solve the task.

4.8. Tasks of a priori unbounded complexity including search in infinite spaces (Pankov *et al.*, 2012; Pankov *et al.*, 2018). The contestant is to reduce the task to search in finite space logically.

4.9. Guessing theorems by the contestant (particularly, for reducing in 4.8). While creation of the task the jury is to prove such theorems strictly but the contestant uses them swiftly. Catching sight of regularities in beginnings of sequences also is effective.

4.10. Discrete tasks of continuous content (Pankov, 2013). They are difficult to solve because the optimal way is mixed: to grope the optimal solution by means of Analysis and to find it by "brute force in small domain".

4.11. Pattern recognition (with strict formulation) (Pankov *et al.*, 2020).

4.12. Real processes executed by 2D- and 3D-printers (with nouns of pixels, voxels, spexels, timexels: space primitives existing during one temporal step) and hypothetical 4D = (3D + time)-printers (Pankov *et al.*, 2021).

4.13. Geometrical tasks for pixels. They have the following preference: numbers of pixels ("area" of any figure or "length" of a "segment") are defined and calculated directly.

4.14. Turkic languages are agglutinating and have strict rules to add affixes. For example, consequence of vowels can be AYAY… or EIEI… or OUA… only; KITEP(book) + DA(locative) + BY(?) = KITEPTEBI(in the book?).
It gives capacities to create tasks on lexicographical order, on numbers of "words" meeting some rules.

4.15. Non-substantiated but practically effective methods (can be used by the contestant if results are seen during the content or the contestant risks). If all or many tests are passed then the jury will not check the text of program. For instance, greedy search with some improvements.

4.16. We do not mean and propose any general algorithms to solve some tasks. On the contrary, we suppose that such algorithms with traditional estimation *O(N...)* in some cases (4.1, 4.3, 4.8, 4.10, 4.11) do not exist and preferences of such tasks are that each task demands its own algorithm, with little discoveries, to smooth out the effect of training contestants.

4.17. To avoid any cribbing, we use parameterized tasks sometimes (Pankov *et al.*, 2015).

5. Examples of Tasks

We cite tasks of various levels, since 2002. Some of tasks are cited non-formally, not completely. We omit the ranges of input and scoring. Also, we use "general tasks" which can be specified according to the level of competition.

We write "input" in examples in abbreviated form: lines are separated with the sign \ .

Task 1 (Horse). Let Lake looks like an isosceles triangle, the basis of the triangle (northern coast) is 190 km and height (width of Lake) is 60 km. Village is located on northern coast of Lake at 20 km from the western corner. Horse runs with speed of 20 km/hour and swims with speed of 10 km/hour. Write a program: A) to show Lake and Village; B) to enable User to show any point on the coast of Lake; C) to draw the fastest way for Horse or D) to show the motion (in scale of 1 hour = 1 sec.) of Horse from this point up to Village along such way.

Comments. 2002 was the year of Horse. The task reflects a historic fact. This village was named after Horse which had crossed the Issyk-Kul lake in XVIII century.

Task 2 (Mice). Given a graph, its vertices are "houses". 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 all 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].

Task 3 (Train). 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 one of *H*.

Task 4 (Snow). Let the streets in the city form a rectangular grid. The firm *Logic* [sponsor] is situated at a given crossing (*X,Y*). Two friends wish to come to *Logic*. 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 path.

Task 5 (Mouse). 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 *X1* and *X2* meters. The men's velocity is 1 meter/second. Write a program calculating the minimal time to catch the mouse in any case.

The following three tasks are specifications of the general task on grammar of Turkic languages.

Task 6 (Vowels). "Words" contain vowels A, E, I, O, U, Y. There must be either consecutive same vowels or the following pairs of consecutive different vowels: AY, YA, EI, IE, OU, UA. Given is a "word" *W* containing more than one vowel. At least how many vowels must be erased from *W* to obtain a new word, the sub-sequence of vowels of which contains only permitted pairs of consecutive vowels?

Example. Input: TOOFEIGUZAEEWYQ Output: 4

Task 7 (Vowels-2). How many words of given length L, made of letters C, A, E, I, O, U, Y having at last one vowel and meeting conditions of Task 6 exist? Output (this number mod 1000).

Task 8 (Vowels-3). Given a word meeting conditions of Task 7. What is its number in lexicographical sequence of all such words with same length? Output (this number mod 1000).

Example. Input: AY Output: 2

Task 9 (3D-printer). The *X*- and *Y*-axes are horizontal, the *Z*-axis is down. Sides of all cubes are equal 1 and parallel to the axes, coordinates of their centers are integer numbers. The lower semi-space "$Z \leq 1$" is filled with cubes. The initial coordinates of Cube-printer are (0, 0, 0). Given one, two or three cubes with coordinates in $[-N, N] \times [-N, N] \times [1, N]$. At each step Cube-printer moves by one along one of axes and erases a met cube. How many steps of Cube-printer are necessary to erase the given cube(s) (with cube(s) over them only)?

Example for two cubes: Input: 2 \ 8 7 1 \ 9 7 5 Output: 21

It is seen that the complexity of this task does not depend on $N > 10$.

Task 10 (Robot). Cubic Robot of volume 1 moves in continuous media (for instance, the warm iron cube moves in dense snow). Its edges are parallel to *X-, Y-, Z*-axes. A "shift" of Robot is its motion along or against one of the axes by an integer number J ($|J| \leq 10^{12}$). Given a sequence of 2..6 shifts, find the volume of Robot's "trace" (empty space in media made by Robot's motions including Robot itself).

Example for three shifts. Input: 3 \ Z 5 \ X –6 \ X 4 Output: 12

Task 11 (Triangle). All numbers are integer; square grid is considered.

Given two points A and B different from the origin of coordinates O (four numbers XA, YA, XB, YB in -1000000..1000000). At each step point B can move along a side or along the diagonal of a cell. To obtain a rectangular triangle OAB how many steps are necessary?

Example: Input: 900000 0 900500 0 Output: 500

Task 12 (Virus). Given the initial position of Virus (integer numbers XV, YV) on an N×N-square grid. At each step you can put an obstacle on a point of grid. In response, Virus tries to step to the neighbor point consequently East; North; West; South. How many steps are necessary to catch Virus?

Example: Input: 1 1 Output: 4

It is seen that the complexity of this task does not depend on $N > 10$.

Task 13 (Sulaiman-mount) [with Museum, in the middle of city of Osh]. Mice are going to celebrate New Year – Spring Equinox on Top of Mount. Now there are M mice and J nuts at Museum and T nuts at Foot of Mount. Time of Foot-Museum movement and Museum-Top one is 10 minutes. One mouse can carry one nut. Find the minimum time (minutes) to deliver all nuts to Top.

Example: Input: M = 3, J = 2, T = 2 Output: 50

General Task 14 (Cell connection). Given some distinct points on a square grid. What is the minimal total length of broken line(s) drawn along sides of cells and connecting all given points?

[Using geometrical proximity yields more effective algorithm than ones for general graph].

By our experience, some who have good command over programming do not know mathematics. Every year we give

General Task 15 (Geometry). Given two figures (rectangles, triangles, segments …) with integer coordinates of vertices (of endpoints …). Find the area of the intersection (of a figure generated by these figures …). Output: P/Q

where P and Q are natural numbers and GCF(P,Q) = 1 (Q may be 1).

There "naturally" arise many special cases in such tasks and contestants spend much time.

Task 16 (Pixels). At first, in an N×N-display boundary pixels are yellow and other pixels are red. If the pixel (X Y) is red then by this command itself and pixels "up, down, left, right" consequently are painted yellow. By the command "W" painting is run until the boundary along each of four directions; by the command "F" painting is run until the first met yellow pixel along each of for directions. After executing of 2..10 such commands how many red domains will turn out?

Example for three commands: Input: 3 \ W 2 4 \ F 3 5 \ W 4 3 Output: 3

It is seen that the complexity of this task does not depend on $N > 100000$.

Task 17 (Roads). Denote points Bishkek(B), Suusamyr Fork(A), Jalal-Abad(J), Talas(T), Osh(O), Karakol(K), Balykchy(L), Naryn(N), Batken(E). Highway distances are: AT = 102; AJ = 377; AB = 193; JO = 106; OE = 240; LB = 179; LK = 216; LN = 180 (km). The speed limit is 60 km/hour. Competition is announced for far apart pairs of drivers. Because of Covid-19, distance (along highway) between them must not be less than 5 km. Now the first driver with car is at X1 point and the second driver with car is at X2 point. The first driver is to reach Y1 point and the second driver is to reach Y2 point ($X1 \neq X2$, $Y1 \neq Y2$), $(X1, X2) \neq (Y1, Y2)$). Find the minimal time (minutes) for it.

Example: Input: J O O J Output: 971

Task 18 (Rectangles). An image is presented at an (white) N × N-display by black pixels. Split the image into the minimal number of (non-overlapping) rectangles. Output this number.

Example ($N = 4$): Input: 0001 \ 0110 \ 1111 \ 0110 Output: 4

Task 19 (Rectangles-2). … Present the image as the union of minimal number of rectangles. Output this number.

Example ($N = 4$): Input: (the same) Output: 3

Task 20 (Embeddings). A natural number N and two words of lengths more than 2 are given. Output a word of length N containing these words as many times as possible, and number of these "embeddings".

If there are some such words then output the first of them in lexicographical order. If there are not such word then output NO 0

Example 1: Input: 8 NZG ZNZ Output: ZNZGZNZG 4 [NZG does 2 times, ZNZ does 2 times]

Example 2: Input: 9 LLL BKTL Output: LLLLLLLLL 7 [LLL does 7 times, BKTL does 0 times]

6. Conclusion

We hope that this paper would diversify the scope of tasks for informatics olympiads, making them more engaging for young people, and attracting contestants' attention to vast applications of informatics, inspire a greater interest of young people in learning sciences and perhaps even in helping to make an appropriate career choice in future. Also, we invite to make acquaintance with tasks of our preceding Olympiads at

`https://olymp.krsu.edu.kg/GeneralProblemset.aspx`

References

Dagienė, V., Skupienė, J. (2007). Contests in programming: quarter century of Lithuanian experience. *Olympiads in Informatics: Country Experiences and Developments*, 1**,** 37–49.

Pankov, P.S., Oruskulov, T.R. (2007). Tasks at Kyrgyzstani Olympiads in Informatics: Experience and Proposals. *Olympiads in Informatics: Country Experiences and Developments*, 1**,** 131–140.

Pankov, P.S. (2008). Naturalness in Tasks for Olympiads in Informatics. *Olympiads in Informatics: Country Experiences and Developments*, 2**,** 16–23.

Pankov, P.S., Baryshnikov, K.A. (2009). Representational Means for Tasks in Informatics. *Olympiads in Informatics*, 3**,** 101–111.

Pankov, P.S. (2010). Real Processes as Sources for Tasks in Informatics. *Olympiads in Informatics*, 4**,** 95–103.

Pankov, P.S., Oruskulov, T.R. (2011) Kyrgyzstan National Report on Olympiads in Informatics. *Olympiads in Informatics,* 5**,** 155–160.

Pankov, P.S., Baryshnikov, K.A. (2011). Tasks of "Mission impossible" and "Mission impeded" types. *Olympiads in Informatics,* 5**,** 113–119.

Pankov, P.S., Baryshnikov, K.A. (2012). Tasks of a Priori Unbounded Complexity. *Olympiads in Informatics*, 6**,** 110–114.

Pankov, P.S. (2013). Tasks in Informatics of Continuous Content. *Olympiads in Informatics*, 7**,** 101–112.

Pankov, P.S., Baryshnikov, K.A. (2014). Tasks in Informatics with Pre-Existing Algorithms. *Olympiads in Informatics*, 8**,** 145–155.
Pankov, P.S., Janalieva, J.R. (2015). Conducting complex competitions in informatics with individual tasks. *Olympiads in Informatics*, 9**,** 163–172.
Pankov, P.S., Kenzhaliev, A.A. (2018). Combinatorial property of sets of boxes in Euclidean spaces and theorems in Olympiad tasks. *Olympiads in Informatics,* 12, 111–117.
Pankov, P.S., Kenzhaliev, A.A. (2020).. Pattern Recognition and Related Topics of Olympiad tasks. *Olympiads in Informatics,* 14, 143–150.
Pankov, P.S., Imanaliev, T.M., Kenzhaliev, A.A. (2021). Automatic Makers as a Source for Olympiad Tasks. *Olympiads in Informatics*, 2021, 15**,** 75–82.
Burton, B. A. Heron, M. (2008). Creating Informatics Olympiad Tasks: Exploring the Black Art. *Olympiads in Informatics: Tasks and Training*, 2, 16–36.
Diks, K., Kubica, M., Radoszewski, J., Stencel, K. (2008). A proposal for a task preparation process. *Olympiads in Informatics: Tasks and Training*, 2**,** 64–74.
Kemkes, G., Cormack, G., Munro, I., Vasiga, T. (2007). New task types at the Canadian computing competition. *Olympiads in Informatics: Country Experiences and Developments*, 1**,** 79–89.

P.S. Pankov (1950), doctor of physics-mathematics sciences, prof., corr. member of Kyrgyzstani National Academy of Sciences (KR NAS), was the chairman of jury of Bishkek City OIs, 1985–2013, of Republican OIs, 1987–2012, participates in National Olympiads since 2020, was the leader of Kyrgyzstani teams at IOIs, 2002–2013, 2018–2021. Graduated from the Kyrgyz State University in 1969, is a head of laboratory of Institute of mathematics of KR NAS.

K.A. Uraiymov (1989), International Educational Institution “Sapat”. Deputy leader at IOIs, 2019–2021. Coach of 9 medalists at IOI, coach of over the 100 medalists of any informatics Olympiads and coach of all winners of National Olympiads in informatics since 2014 until present. Graduated from the Kyrgyz Tukish “Manas” University in 2010.

A.A. Belyaev (1978), Kyrgyz-Russian Slavic University. Deputy leader at IOI’2018. Regional Director of NERC ICPC Kyrgyzstan Regionals.