Olympiads in Informatics, 2013, Vol. 7, 78–89



# Latvian Olympiad in Informatics – Lessons Learned

Mārtiņš OPMANIS
*Institute of Mathematics and Computer Science, University of Latvia*
*Raina Boulevard, LV-1459 Riga, Latvia*
*e-mail: martins.opmanis@lumii*

**Abstract.** The paper gives an insight in the format of annual (since 1988) Latvian programming competition for secondary school students – Latvian Olympiad in Informatics (LIO) – as well as describing findings and drawbacks of the current, 26th LIO. As an illustration a couple of task examples are given. Problems concerning grading process are discussed.

**Key words:** olympiads in informatics, competition tasks, grading, jury responsibility.

## 1. Introduction

The introduction of computers in Latvian schools started in the middle of 1980s as an ex-USSR program of school computerization. After the restoration of independence of Latvia in 1990 this process got more power and after joining the European Union in 2004 several EU projects were also devoted to updating the ICT equipment in schools and educating teachers. According to the newest European Commission survey (EC, 2013), in the benchmarking group "digitally equipped school", Latvian schools are ranked over the EU average having also very good Internet availability and speed marks. Today, two ICT related disciplines are taught in Latvian schools – the mandatory discipline "Applied Informatics" (starting with grade 5) and a mandatory "Basics of Programming" (for grades 10–12). Often, the naming of these disciplines is a source of confusion, because in the early years of computers, in schools, there was just one discipline "Informatics", usually used as a synonym for "Programming". Even now statisticians do not try to distinguish between these disciplines and from the national statistics registers (Latvian Statistics) you can get only the total number of school students attending these two disciplines: 84,802 in the season of 2011/2012 (53 222 in grades 1–9, and 31 580 in grades 10–12).

Teaching algorithms and programming is an essential part of the ICT education process and now the process is started to improve the teaching of algorithms (or "bring them back") in Latvian schools. To encourage teaching of programming, since the very beginning of computerization era, algorithic programming competitions have been organized. Traditionally, at that time, the competitions for high school students in disciplines like as mathematics and physics, with unusual tasks going beyond the ordinary school programme, were called "olympiads".

The first programming competition in Latvia which can be named "olympiad" was organized by young scientists of Computing Centre of Latvian State University (now Institute of Mathematics and Computer Science, University of Latvia) in February, 22nd 1986. Roots of this competition go back directly to the famous Latvian mathematics olympiads conducted at that time by professor Agnis Andžāns (Ramāna and Andžāns, 2002). Prof. Andžāns also participated in setting up of programming (or, in the early years more appropriate name would be – algorithmic) olympiads.

Formally the Latvian Olympiads in Informatics (*Latvijas informātikas* (*programmēšanas*) *olimpiāde*, LIO) started in 1988. Today the LIO together with olympiads in mathematics, physics, chemistry, geography and biology is one of 6 "big"(having a corresponding international olympiad) science olympiads held annually. In the season of 2012./2013 the 26th LIO took place (`http://www.lio.lv`).

The LIO is the most popular individual programming competition in Latvia. Besides the LIO regular team competitions, like the team competitions in mathematics and informatics "Ugāle" (Opmanis, 2006) and Lattelecom IT Olympiad (`http://olimpiade.lattelecom.lv/olimpiade2011/`) are organized.

Winners of the LIO participate successfully in the International Olympiads in Informatics (IOI; `ioinformatics.org`) and their achievements were recognized by the authors of an international survey (Nedkov, 2012).

## 2. Format of the LIO

In the history of the LIO we can find essentially different formats – in the early days it was a one day theoretical pen-and-paper algorithm design and analysis round followed by a practical round on computers. Later, the format of two rounds of different lengths (five and four hours) was used, and smaller differences were introduced – like different maximum points for a solved problem. Current the LIOs are strongly connected to such international events as Baltic Olympiad in Informatics (BOI; Bulotaite, 1997; Poranen *et al.*, 2009) and the IOI and they try to follow the standards of these competitions.

Today the LIO is a four level programming competition. Competitions of all levels are graded by using an automated grading system. The first level – school competition – is aimed at newcomers but is open as well for all students willing to brush up their competition participant skills. There are two divisions – junior division (grades 8–10) and senior division (grades 11–12). For each division three tasks are prepared and it is supposed that tasks must be solved in the standard time for a single olympiad round – 5 hours. Participation in the school level competition is compulsory – it has no qualifying meaning. It is intended rather for testing further competitor willingness to participate in the next years' LIOs. It must be mentioned that despite the formal lower bound of grade 8, younger participants are allowed to participate as well. The youngest participant at this year's LIO was from grade 5. Also a lot of interested teachers take part as contestants in first level competitions.

The second level are regional competitions which have qualifying status and the main goal of participants is to qualify for the next level competition. Since 2006 when LIO's jury was seriously accused for incorrect grading, all LIO procedures were described at a very detailed level and nowadays the text of the LIO Regulations is approximately four times longer than regulations of any other Latvian scientific olympiad. Such strong regulations give no way for manoeuvres even if some of contestants are not able to participate at regional level. If the student due to any reason could not participate at the regional level competition and did not win a medal year before (and therefore qualifies for competition), there is no way to get to the next stage and therefore road to the IOI this year is closed.

Second level competitions are organized in one round, they have the same two age divisions as the first level and there are also three tasks given for five hours. Competitions are organized across the entirety of Latvia in a so called "supervised online" way. This means that all competitors from each region come together in a particular school and compete under the supervision of a teacher responsible for the competition in the region. The real situation in Latvia is such that regional competitions *de facto* are organized only in places where there are enthusiastic teachers willing to spend their free time for such extra-curricular activities. It seems, there are no other serious factors(such as population of a particular city) influencing popularity of programming and participation rate in the LIO. At the regional level grading is done by the central grading system and so all the participants despite their geographical distance are in virtually almost the same contest situation. This year, the number of contestants was lower than in the previous year – 66 in the junior division (78 at 2011) and 96 in the senior division (130 at 2011). Only representatives of 41 secondary school (out of 835 having Internet access) participated in the second level competition.

For this year, because of legal matters it was strictly stated that no more than 40 participants from each group would qualify to the next level competition. These participants were to be chosen according to four criteria (in the given order):

1. participant's result in his/her region is the best and he/she is the only one in the region with such a result, and it is at least 25% from the 10 best overall participants average result;
2. previous year LIO medalist;
3. previous year Baltic or international OI participant;
4. best participants according to their overall results not qualified according to the previous criteria.

The third level – country competition – is sometimes also called the "finals" and it is organized as an on-site competition. In two consecutive days two five-hour rounds with three tasks each are organized. Age groups are the same as in the previous levels. Already for the fifth year in a row the LIO finals were organized outside the capital of Latvia – Riga. This tradition started in 2009 following a similar pattern used in the neighbour country Lithuania. In these five years, four times the competition was organized in the local universities and university colleges (Ventspils, Rēzekne, Jelgava, Daugavpils) and one time in a state gymnasium (Cēsis). This experience was highly positive, it showed that outside of Riga the event becomes a real festival and local organizers are even better (definitely because of a willingness to demonstrate their ability). Again, because of legal issues this year it was the first one when for all Latvian scientific olympiads a strict medal allocation algorithm was established. The main idea comes from the IOI medal allocation algorithm. It is stated that at least 1/24 of all participants will be awarded with gold medals, at least 1/8 of all participants with gold or silver medals and at most 1/4 of all participants with any of the medals. Such an algorithm allows knowledge in advance of the number of medals required (there can be some deviations only due to equivalent numbers of points received).

After the third level competition all medal winners are invited to the fourth level competition – the selection round for the participation in Baltic OI. The selection round is organized as an on-site two consecutive days competition in a single age group. The six best participants are included in the Latvian team for participation in Baltic OI. Interestingly, in this year the three best participants from the youngest group of the country competition qualified for Baltic OI showing the first, second and fifth result in the selection round.

## 3. Lessons Learned from the Current LIO's Country Competition

Despite almost the same structure year-by-year, the previous experience is analyzed every year and some innovations are tried.

### 3.1. *Introducing "First Subtask"*

An important innovation at the State level competition was introduced in order to treat the so called "0-frustration" (i.e., scoring of 0 points at the State level competition) of contestants and their teachers. From time to time this issue was raised during discussions, more or less openly blaming the problem setters for making the competition "too hard". From the viewpoint of organizers, it was not always clear whether gaining 0 points is caused by technical problems like impossibility to technically submit solution to the grading system, a total misunderstanding of the task or the complexity level of the task being really too high.

For this year's competition it was decided to add into every task description a special part – the so called "first subtask". Namely, for one or several test cases there were given exact input data. To get points for this subtask (2 points out of the maximum possible 100) it was enough to solve these particular test cases on a paper or by using simple tools available on contestant's computer, and after this write a simple program which solves these test cases only.

The intention of this invention was distinguishing clearly between contestants who did not understood the idea of the task at all from that the ones who cannot solve the task because of its hardness. The experiment was successful – for the first time, there were no participants with 0 scores at all in both age groups. However, instead of 12 points which could be collected theoretically in such an easy way during two competition days, there were still contestants scoring just 2 or 4 points. This clearly shows that the "first subtask" approach should be continued and the exact reasons of the weak performance of particular students must be studied.

### 3.2. *Jury Mistake(s)*

Following the last years' IOI tradition at LIO partial feedback is also provided – for every accepted submission there is a possibility for several tests see the exact result of grading, whether it is correct or not. From the early years of innovation of particular or full feedback the author warned the IOI community about the impossibility of correcting jury mistakes in case of erroneous grading during contest. However, till now such warnings were not heard.

Taking in account the author's sceptical attitude to using feedback in competitions, it is more ironic, that the situation which can be easy defined as *force majeure* took place in the competition conducted by the author as head of the jury.

The rationale of the case was the following: on the first competition day one of the tasks needed a checker for grading due to non-unique correct results. During the competition the checker was run with incorrect parameters using a file with correct jury solution instead of a file with actual contestant's solution. This led to incorrect grading of huge number of solutions because all solutions which fitted in the time limits and ended with normal exit code were accepted and even more – it was reported back to contestant that all "visible" task groups were solved correctly.

Afterwards it was quite interestingly to analyze how contestants proceed during competition and afterwards. It was obvious, that most of contestants receiving message "all is correct" stopped working on this particular task. Clear psychological parallels with computer games can be seen there – if a level of the game is completed, you proceed to the next one and do not bother whether you were smart or just lucky. The dangerous part is that among these participants were also such participants whose programs were so trivial that they simply could not be correct! In general a serious alarm signal is that contestants take too much account of the grading system and are too lazy or too weak in testing. Even more – full or even partial feedback stimulates contestants not to test their programs which was definitely not the case without feedback. One way to cope with this would be to give the contestants the possibility of creating an appropriate test case and checking whether execution result is the same for contestants and jury programs. Then level of program testing will directly depend on the content of test case created by the contestant. Further these tests may also be used in the full test set as suggested before (Opmanis, 2006).

But let us come back to story about the jury mistake. After the competition round the jury got no official protests. From different sources came unofficial rumours that several contestants have decided that the "grading system is broken" because full score were given for very weak programs and intuitively contestants felt that "something is wrong". After getting even such unofficial impulses, the jury quite fast recovered the source of problem and regraded all submissions of that particular task. Regrading influenced 23 out of 36 contestants lowering their results (in some cases from 100 to 0 points). Regrading was the only reasonable solution seen by jury and corresponds to previous practice at OI. However, adult participants of the LIO were not united in support of the jury decision – other options like exclude results for this particular task or even all competition round were mentioned. One more theoretically possible option is "leave results as is" – so "keeping promise about information submitted during contest".

This uncomfortable situation predicted in theory, also in practice showed its very ugly face – there simply was no fair-for-all decision. If you try to analyze benefits and drawbacks, you can easy see that always some contestant can be placed in "unfair" position. I would like to emphasize that this took place in a situation with particular feedback. The author's feeling is that in case of full feedback things will be even worse, because the feeling "all is correct" must be even stronger. And if there is willingness to keep feedback in its current form, there simply must be some paragraph in the rules – what to do in the cases of jury mistake that are found after the competition round.

Besides the problem that contestants are not testing their solutions and, in the worst scenario, are losing this essential ability necessary for software developers, this episode also raises the question about the role of the grading system. Some authors argue that grading system is just a simple technical tool like a stop-watch in the field and track competitions. This is almost true for the "old" approach where results of main grading were not available during contest. In the case of immediate feedback the grading system acts more like a referee in sports and therefore is a serious player at the playground. If we try to follow this analogy and find out what is said about referee's (or more general, "officials") errors and mistakes in the rules for various sports, we find a lot of common together with several quite different principles. For example, in football (`http://www.fifa.com/aboutfifa/footballdevelopment/technicalsupport/refereeing/laws-of-the-game/index.html`) referee has the authority to decide on all points and his decision is final. There is no way how to influence referee's decision even when it is wrong. In the basketball rules (`http://www.fiba.com`) there is section named "Correctable errors" and similar section "Correcting errors" is in tennis rules (`http://www.itftennis.com/officiating/rulebooks/rules-of-tennis.aspx`). In these sections a limited number of possible judging errors is listed and clear algorithms how to proceed are given. In ice hockey rules there are guidelines for simple situations like "In cases of an obvious error in awarding a goal or an assist which has been announced, it should be corrected promptly" (`http://www.iihf.com/iihf-home/sport/iihf-rule-book.html`). Quite obvious, that there may be also errors not listed in the rules. In the tennis rules error situations are mentioned where the decision whether to correct this error or not depends on the moment when the error is discovered. In basketball and ice hockey there are clear deadlines like signing a game protocol after what no errors can be corrected at all.

Trying to use similar principles in the programming contests the question must be answered how to proceed in case when a grading system error is found after the competition round. Ask contestants to repeat their performance? Cancel the competition results? At the programming competitions the possibility of running into problems with testing is much more higher than in sports because of the complexity of grading systems and it would be wise to be prepared for such situations. In general, the absence of an overall good analogy with sports is one of the reasons why it is hard to obtain clear and relatively simple procedure for the programming competitions.

At the end I would like to express my strong feeling that the philosophical questions like as "what exactly are the goals and contents of the competition", "does full feedback change the competition paradigm" must be answered taking in account current situation, and as soon as possible. As a source for thinking could serve the discussion raised after the "CodeForces" competition and trying to understand whether it is acceptable to change tests after seeing contestant submissions (Mitrichev, 2013).

## 4. Task Examples

To give insight into LIO tasks, let us discuss two examples of tasks – "Valid booklets" (authors – me and Rihards Opmanis) and "Benefit" (author Sergejs MeIniks) from LIO selection round. The task "Valid booklets" was proposed for Baltic OI in 2011, and the task "Benefit" – for IOI 2012. However, both tasks were rejected by the respective scientific committees. The task "Valid booklets" is interesting because its solution uses the pigeonhole (or Dirichlet) principle present in IOI Syllabus (The International Olympiad in Informatics Syllabus, 2013) but not very often really used in task solutions. For the national competition, the story part of the task "Benefit" was completely reformulated to hide clues for the suggested model solution. Below, both task descriptions are slightly modified for publication omitting technical information about formats of input and output data. a mathematical style of solution description is chosen to allow the reader to follow the reasoning better.

### 4.1. *Task "Valid Booklets"*

#### 4.1.1. *Task Description*

An exam paper consists of $K$ pages numbered consecutively: $123\dots K$. There were made $M$ copies and all pages placed in one big pile: $1\dots K1\dots K1\dots K$ ($M$ times). Using a booklet stitching machine there were created booklets taking pages from the pile. Each booklet (maybe except the last one) contains $N$ pages.

We will say that booklet is *valid* if it contains full set of exam paper pages in the right order (pages $1\dots K$ consecutively).

Write a program which for given values of $K$ (the number of exam paper pages), $M$ (the number of copies) and $N$ (the number of booklets, $N \leqslant 9 \times 10^{18}$) calculates number of valid booklets. It is known that $K \times M \leqslant 9 \times 10^{18}$. See Table 1.

#### 4.1.2. *Solution*

Let's start with several observations:

1. if $N < K$, then the result is 0,
2. if $N = K$, then the result is M,
3. if $N \geqslant 2K - 1$, then each full group definitely is valid booklet, because there always is a group of $K$ consecutive pages starting with the first page. The number of booklets therefore is. The last booklet with less than $N$ pages can add one more valid booklet if.

Table 1

Examples

| Input data | Output data | Comment |
|---|---|---|
| 2 6 3 | 4 | The following booklets were created (exam paper sets are underlined): (1 2 1) (2 1 2) (1 2 1) (2 1 2) |
| 4 6 5 | 3 | The following booklets were created (exam paper sets are underlined): (1 2 3 4 1) (2 3 4 1 2) (3 4 1 2 3) (4 1 2 3 4) (1 2 3 4) Despite the fact that the last booklet contains only four pages, it still contains full set of exam paper pages and therefore is valid. |

The only remaining case is $K < N \leqslant 2K - 2$. According to pigeonhole principle there are one or two first pages in each full group. a booklet is valid if there are two first pages or group ends with page number $K$ (only one of these two situations can take place).

The number of first pages located in full groups is
$N_{FP} = \begin{cases} M, & if\{\frac{KM}{N}\} < K, \\ M-1, & if\{\frac{KM}{N}\} \geqslant K. \end{cases}$.

The number of full groups containing two first pages can be calculated as
$N_{FP} - [\frac{KM}{N}]$.

The number of full groups ending with page number $K$ can be calculated as $[\frac{KM}{LCM(K,N)}]$, where $LCM(K,N)$ is the least common multiple of $K$ and $N$. Using the formula $KN = LCM(K,N) \times GCD(K,N)$ where $GCD(K,N)$ is the greatest common divisor of $K$ and $N$, the previous formula can be transformed as follows:

$$\left[\frac{KM}{LCM(K,N)}\right] = \left[\frac{M \times GCD(K,N)}{N}\right] = \left[\frac{M}{\left[\frac{N}{GCD(K,N)}\right]}\right].$$

This transformation may be helpful to limit the intermediate results.

An uncompleted group can still be valid booklet if $\{\frac{KM}{N}\} \geqslant K$.

Summing up all these calculations we obtain a surprisingly simple formula:

$$M - \left[\frac{KM}{N}\right] + \left[\frac{M}{\left[\frac{N}{GCD(K,N)}\right]}\right]$$

4.2. *Task "Benefit"*

4.2.1. *Task Description*

Given $N$ cards, having one side coloured in green and the other side in a red colour. On each side some integer is written. When any two cards are chosen (lets denote them by

Table 2

Example

| Input data | Output data | Comment |
|---|---|---|
| **5** | **114** | The maximum difference of benefits is |
| **9 –1** | | between the fourth and the second card: |
| **7 8** | | $9 \times 8 - 7{\times}(-6) = 114$ |
| **–2 4** | | |
| **9 –6** | | |
| **3 5** | | |

$A$ and $B$), it is possible to calculate *the benefit of $A$ against $B$*. This benefit is calculated as the product of the numbers written on the green side of card $A$ and on the red side of card $B$.

For example, if 10 is written on the green side of the card $A$, and 3 on its red side, and 7 is written on the green side of the card $B$, and $(-2)$ on the red side, then the benefit of $A$ against $B$ is $10 \times (-2) = -20$ and the benefit of $B$ against a is $7 \times 3 = 21$. When both benefits are calculated, it is possible to find difference between these benefits. In the given example it is $-41$. If the same cards would be chosen in the reverse order, the difference of benefits would be 41.

Write a program, which, for a given description of $N$ ($N \leqslant 2 \times 10^5$) cards finds the maximum possible difference of benefits. It is known that all the numbers written on cards are between $-2 \times 10^9$ and $2 \times 10^9$. See Table 2.

4.2.2. *Task Solution (by Sergejs Meļņiks)*

We begin with some observations that reveal the geometric meaning of the task. Let us denote by $s_{ij}$ the benefit of card with index $i$ against the card with index $j$, and by $x_i$ and $y_i$ – the numbers written on the green and red sides of the card with index $i$ respectively. We can observe that:

1) $s_{ij} = -s_{ji}$, i.e., the maximum cannot be negative, because for any negative $s_{ij}$ we can choose $s_{ji} = -s_{ij} > s_{ij}$ Therefore, it suffices to find the maximum value of $|s_{ij}| = |x_i y_j - x_j y_i|$.
2) Replacing $x_i$ by $-x_i$ and $y_i$ by $-y_i$ does not change the value of $|s_{ij}| = |x_i y_j - x_j y_i| = |-(x_i y_j - x_j y_i)| = |(-x_i) y_j - x_j(-y_i)|$.
3) Consider a triangle OAB in the coordinate plane, where O – the point of origin having coordinates (0,0), A – the point with coordinates$(x_i, y_i)$ and B – the point with coordinates $(x_j, y_j)$, then $|x_i y_j - x_j y_i|$ equals to doubled area of the triangle OAB.

Therefore, the original task can be reduced to the following one:

Let us construct the set of points $P = \{P_0, P_1, \ldots, P_{N-1}, P_N\}$, where the coordinates of $P_0$ are (0,0), and for all $i = 1, \ldots, N$ the coordinates of $P_i$ correspond to the "card" $(x_i, y_i)$ as follows:

$$\begin{cases} (-x_i, -y_i) & \text{if } y_i < 0 \text{ or } (y_i = 0 \text{ and } x_i < 0), \\ (x_i, y_i) & \text{otherwise.} \end{cases}$$

The triangle must be found with the largest area from all triangles such that one vertex is the point $P_0$ and two other vertices are placed in the two different points $P_i$ and $P_j$.

Let us denote the convex hull of $P$ by $\mathrm{conv}(P)$. By construction, $P_0$ belongs to $\mathrm{conv}(P)$ because $P_0$ is the leftmost of all the lowest points of $P$.

By purely geometrical means we can prove that all vertices of the desired triangle with maximum area are located in the vertices of $\mathrm{conv}(P)$.

Let the points $A, B, C$ be located in the interior of a convex polygon $M$.

We want to construct a triangle $ABV$ such that $V$ is the vertex of $M$, and the area of $ABV S_{ABV} > S_{ABC}$.

The ray $AC$ crosses the boundary of the polygon $M$ at some point $D$. Obviously, $S_{ABD} > S_{ABC}$.

If $D$ is a vertex of $M$, we set $V = D$. Otherwise, $D$ is located on the some side $UW$ of $M$ (see Fig.1). If $UW$ is parallel to $AB$, then $S_{ABU} = S_{ABW} = S_{ABD} > S_{ABC}$, otherwise one of the points $U$ and $W$(let this be point $W$, for definiteness) is located closer to $AB$ than the point $D$, and the other vertex (point $U$) is located further from the $AB$ than point D. Then, choosing $V = U$we obtain that $S_{ABV} > S_{ABC}$.

Now we can extend the ray $VB$ till the intersection with the perimeter of the polygon $M$and use the same reasoning as above. Afterwards, in the same way we can extend the ray $VA$.

To construct $\mathrm{conv}(P)$ let's sort all points $P_i$ except $P_0$ by the polar angle and then apply the Graham scan. Note that all necessary comparisons can be easily implemented in integer arithmetic. The time complexity of constructing the convex hull is $O(N \log N)$ with required memory is linear in $N$.

Let $\mathrm{conv}(P)$ to be the polygon $B_0 B_1 \ldots B_K$, where $B_0$is the point of origin $(0, 0)$.

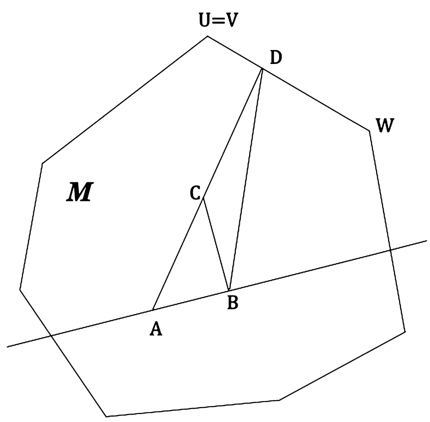


Fig. 1. Triangle with maximum area within polygon.

Let denote by $q(i)$ the number $j$ such that $j > i$ and $S_{B_0 B_i B_j} = \max_{i<m\leqslant K} S_{B_0 B_i B_m}$.
It can be proved that $q(i)$ is a non-decreasing function: if $i_1 < i_2$ then $q(i_1) \leqslant q(i_2)$.
These considerations allow us to find a triangle with the maximum area via a single traversal of the convex hull by consecutive processing related values $i$ and $q(i)$.

## 5. Conclusions

During its 26 years, the idea of Latvian Olympiad in Informatics has shown its vitality. However, changes made since the early years show that there is always space for new ideas and improvements in wide range of directions.

**Acknowledgements.** I would like to thank prof. Kārlis Podnieks for his valuable comments.

## References

Bulotaite, J., Diks, K., Opmanis, M., Prank, R. (1997). *Baltic Olympiads in Informatics*. Institute of Mathematics and Informatics, Vilnius.
EC (2013). *Survey of Schools: ICT in Education, Final Report*. `http://ec.europa.eu/digital-agenda/en/news/survey-schools-ict-education`
*International Olympiad in Informatics*. `http://www.ioinformatics.org`
*Lattelecom IT Olympiad*. `http://olimpiade.lattelecom.lv/olimpiade2011/`
*Latvian Olympiads in Informatics*. `http://www.lio.lv`
*Latvian Statistics*. Databases of the Central Department of Statistics. `http://data.csb.gov.lv/DATABASE/Iedzsoc/Ikgadējie statistikas dati/Izglītība unzinātne/ Izglītība un zinātne.asp`
Mitrichev, P. (2013). *blog at CodeForces*. `http://codeforces.ru/blog/entry/6928`
Nedkov, P., (2012). Young talent in informatics. In: *An AICA – IT STAR Survey, International Conference on Young Talents and the Digital Future*, Milan. `http://www.aicanet.it/eventicontestuali/allegati-eventi-2012/IOI_ Survey-final.pdf`
Opmanis, M. (2006). Some ways to improve olympiads in informatics. *Informatics in Education*, 5(1), 113–124.
Opmanis, M. (2009). Team competition in mathematics and informatics "Ugāle" – finding new task types. *Olympiads in Informatics*, 3, 80–100.
Poranen, T., Dagiene, V., Eldhuset, Å., Hyyrö, H., Kubica, M., Laaksonen, A., Opmanis, M., Pohl, W., Skūpiene, J., Söderjhelm, P., Truu, A. (2009). Baltic olympiads in informatics: challenges for training together. *Olympiads in Informatics*, 3, 112–131.
Ramāna, L., Andžāns, A. (2002). Advanced mathematical education in Latvia. Creativity in mathematics education and the education of the gifted students. In: Andžāns, A., Meissner, H. (Eds.) *Proceedings of the International Conference*, Riga, University of Latvia.
*The International Olympiad in Informatics Syllabus* (2013). `http://people.ksp.sk/~misof/ioi-syllabus/ioi-syllabus.pdf`

**M. Opmanis** is researcher at the Institute of Mathematics and Computer Science of University of Latvia. Since the very first Latvian Olympiads in Informatics in 1986 he is involved in its organization at various positions (last years – head of jury). He is deputy or team leader of Latvian IOI teams since 1996 and team leader of Latvian team at Baltic Olympiads in Informatics since the first Olympiad in 1995 (except 2008 and 2010). M. Opmanis was head of jury of Baltic Olympiad in Informatics at BOI 1996, 1999, 2004 and 2012. Since 2012 he is member of IOI International Committee.