Olympiads in Informatics, 2016, Vol. 10, Special Issue, 25–30

DOI: 10.15388/ioi.2016.special.04

Preparing to Olympiads in Informatics in Tatarstan Republic, Russia. The Experience of Kazan Federal University

Ravil HADIEV, Kamil KHADIEV
Institute of Computational Mathematics and Information Technologies
Kazan Federal University
Kremlevskaya str, 18
420008 Tatarstan Republic, Kazan, Russia
e-mail: rawil.hadiev@kpfu.ru, kamilhadi@gmail.com

Abstract. In the paper we describe the part of the history of Olympiads in informatics in the Tatarstan Republic, Russia which connected with the Kazan Federal University. It began in the 80s of XX century and now there are several centres of Olympiads in informatics in Tatarstan and one of them is the Kazan Federal University.

In the second part of the paper we describe the current approach to organize the lessons on informatics. The main idea is a seamless transition from "lessons 1.0"(teacher prepares all materials for students) to "lessons 2.0"(students prepare all materials for students), like the transition from Web 1.0 to Web 2.0.

Keywords: training activities, camps, national Olympiad in informatics, history.

1. Introduction

At the end of 60s students of high schools of Tatarstan had first lessons on Programming and Informatics. In the early 80s they began to study at Universities and work at companies using computers. In the middle 80s, there was a period when our schools got personal computers.

In 1985 teachers of Computational Mathematics and Cybernetics (CMC) department of Kazan State University (now it is Institute of Computational Mathematics and Information Technologies of Kazan Federal University) organized a first Olympiad in Informatics for Kazan high school students. There were about 40 students in the competition, and its jury chairman was assistant lecture of CMC department Ravil Hadiev. Additionally, in 1986 same staffs of the CMC department of KFU organize first camp in Informatics.

At the same time three factories (Kazan factory of Computers, Elecon and Kamaz with FORT "Dialog" company) began to produce personal computers. In 1987 FORT "Dialog" company suggested to organize the first regional Olympiads in Informatics for students of high schools in Brezhnev (Naberezhnye Chelny) city. And staffs of CMC department of KFU prepared problems for this contest. The winner of this competition was Ziganshin M., a student of one of Brezhnev city's schools. All winners who graduated school that year entered Kazan State University (KFU) and helped to organize next regional Olympiads in Informatics in 1988. That year new winner was a Kuyanov Maxim student of the Kazan school no.131.

At the same time the first regional Olympiads for village schools of Tatarstan was organized and staffs and students of the CMC department of KFU also prepared problems and took part in organizing work.

Since 1990 The Ministry of Education of Tatatrstan has been organizing regional Olympiads in Informatics using own staffs.

The experience of this work was published in (Hadiev *et al.*, 1992) by Hadiev R.M., Anufriyeva A.I., Suleymanov D.Sh. In 1992.

Since 1989 training camps on programming and informatics have been organized. First of them was in Bolshiye Nirsi village, which was organized by staffs and students of the CMC department of KFU. Later using same ideas was organized other camps like "Selet" and "Baytik". First "Selet" camps were organized by same staffs.

Since 2002 staffs of CMC department of KFU have worked with students of university and schools and prepare them for Olympiads on Informatics and contests like ACM ICPC.

2. An Organization of Training for Olympiads in Informatics and Programming Contests with Students in Center

Training process contains several points:

- Weekly theoretical classes.
- Weekly practical classes.
- Open contests, three in a year.
- Two exam contests in a year.
- Training camps.

Students have one theoretical and one practical lesson a week. Students of a school and students of university work together. This situation is useful for both groups of students. On the one hand, it is a competitive atmosphere for university students. They look at smart school students and try to be better. On the other hand, school students do not think that end of school is the end of Olympiads, they see many opportunities for students who interested in programming contests. As a result, school students taking part in contests during the last years of studying at school do not stop and continue to compete at the university. Smart school students look at smart university students that is why do not stop evolution (Hadiev and Khamizova, 2003).

We separate all pupils to five groups:

- Elementary group.
- Beginners group.
- Base group.
- Advanced group.
- Professional group.

Main parameter of separation is knowledge, but it is not study year.

2.1. *Weekly Lessons*

Each group had two lessons a week. One of them is theoretical and the second one is practical.

Theoretical lesson is not lecturing, it is a discussion about how to solve classical problems. Typical scheme is following: teacher gives classical problem. Students try to solve it. If they cannot do it, then the teacher gives little hint, for helping students. If it is not enough, then the teacher gives a second hint and wait again. He gives hints until students do not get the solution themselves. Student have to prove their solutions. Then the teacher tells a classical solution and after that compares two programs. Main parameters for comparison are time and space complexity, easy code, clear code. After discussion of solutions, we debate about about the general approach to solve such kind of problems. We believe that this discussion helps to understand their solutions and will help to solve this kind of problems in future. Sometimes the general approach to solve problems is one type of hints.

This way of giving theory allows to show the ways of problems solving. Both epy proof and finding solution themselves are helping to this goal. We believe that students cannot know all main topics of informatics and programming, but should have ability to solve new problem (it is not important is problem classical or not). Additionally a discussion about general approach is one of the most important parts of the lesson, because most of students can solve some problems but do not understand how to use this solution in the other similar problem.

Rarely we try to make our lectures as a sequence of little problems, which are solved by the way as discussed above. If we cannot do it we ask students to prepare and make a talk, teacher helps them if they need. This approach has two profit: firstly, a student who prepares a talk better understand the topic; secondly, students better understand younger student, but not old teacher.

Practical lesson is a simple online contest with problems on the topics which was learned on previous theory lesson. It gives them practice of contests and short time for solving problems.

The problems for such trainings are prepared by students of higher group (students of base group prepare problems for beginners group, students of advanced group prepare problems for base group etc.). A student who prepares problems gets skills on generating tests, stress solutions, writing statements and of course solve the problems on topics

which they already know. On the one hand it allows to get the skills which helps on the contest (like tests, stress solutions and understanding statements). On the other hand it helps to remind old topics, which make their knowledge stronger.

There is no homework except up solving the problems which didn't solve on practical lesson. Also, we stimulate students to solve problems on different websites like `www.topcoder.com`, `www.codeforces.com`. This kind of "homework" is not hard work at home, but funny taking part in contests. Also, students try to up their rating on web contesters and this is additional fun. We believe that this "homework" is better, because without marks there are no motivation for students except fun and a sense of competition.

2.2. *About Groups*

Elementary group. There are only school students, who do not know programming. Here we learn only simple topics on programming language like "variables", "arrays", "cycles", "functions", but we use "olympic" problems for the lessons. University students do not follow this group, because typically they have programming lessons in their university study program.

Following groups learn algorithms, data structures and problems, which students can see in the contests. We use "spiral" strategy. It means students of the beginners group learn all main topics like algebra, text search, dynamic programming, backtracking, greedy algorithms, graph theory, geometry, data structures but simple algorithms. Students of the beginners group learn same topics and some additional, but base algorithms. Students of advanced group learn advanced algorithms of the same topics and some additional, etc.

Beginners group. In this group students have about 2–3 hour theoretical lesson and 2 hour practical lesson. Here they study simple greedy and backtracking algorithms, work with sequences, prime numbers, etcų, syntactic analysis, KMP, sorting, simple geometry problems, convex hull, BFS, DFS, Hamilton cycle, Euler cycle, simple data structures

Base group. In this group students have about 2 hours theoretical lessons and 2–3 hour practical lesson. Here they study segment trees, Binary search, LCA, phi-function, z-function, scan-line, hash, combinatorics, 3d geometry, meet-in-the-middle, Flow networks, Matching problems and others.

Advanced group. In this group students have about 2 hours theoretical lessons and 3–4 hour practical lesson. Here they learn advanced data structures and algorithms, like persistent data structures, Hungarian algorithm, integrals, probability theory problems, Numerical analysis problems, Group theory problems and others.

Professional group. In this group students have about monthly 2 hours theoretical lessons and 4–5 hour practical lesson once or twice a week. Here they solve different contests and just discuss solutions to difficult problems.

2.3. *Contests*

We organize exam contests at the end of each semester for each group. We use classical problems with little bit changes. The goal of these contests using exact algorithms, which was learned in that semester.

Additionally, we have three open online contest a year. These contests also have two or three divisions. This is not classical problems, but this is problems exactly on topics which was in the previous study period. This competition allows to train in using new knowledge in real contexts. Moreover, this contest is open and our students compete with external people.

Problems for this contest are also prepared by students from higher groups. This is an additional opportunity to remind topics from previous group.

2.4. *Training Camps*

We organize two types of training camps:

- **Training camps at the university.** Students just come to university and have a lesson. Often it is five days training. Every day they solve 5 hour contest and then they listen solution of contest's problems and some theory.
- **Training camps.** Students live in camp and have a lesson. Often it is a two week camp. Every day they solve 5 hour contest and then they listen about the solution of contest's problems and theory on next days.

Second type of camp is better, because they get more information and have less distractions. But we can organize it only once a year (often it is summer). First one less effective, but we can organize it two or three times a year.

Additionally, our students visit other training camps.

2.5. *Future Lessons*

We are planning to add new types of lessons "Hacks solutions". You can hack the solution on contesters like topcoder.com and codeforces.com. This kind of lessons allows to learn how to find mistakes in the solution and find how to fix it. It is good skills ant it is too hard to train them in regular contests.

3. Conclusion

We have big experience in teaching students to solve problems of programming contests. But in same time current situation in this area is changed very fast. On the one hand,

there are new kinds of tools and new kind of opportunities. On the other hand level and knowledge of contestants become higher from day to day. In this context, we try to use new teaching techniques and sachems. We describe our current technique which we have been using for five years and we extend it year by year with new items, but at the same time it has already given good results.

Our main point is organizing of the process "students teach students". Like in web transition from Web 1.0 to Web 2.0. It means that users make content, but not authors of web site (Codeforces, n.d.; topcoder, n.d.; KPFU, n.d.). Similar students form the content of lessons and train other students. We call this process as transitions from "lesson 1.0" to "lesson 2.0". According to this concept teacher is converted from lecture teacher to helper or manager.

References

Codeforces (n.d). *Contester System "Codeforces"*. `http://codeforces.com`

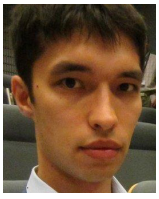
Hadiev, R.M., Anufrieva, A.I., Sulejmanov, D.Sh. (1992). *Workshops on School Informatics: Study Letters for Teachers and Students*. Kazan, Publishing house of Kazan University.

Hadiev, R.M., Khamizova, E.Sh. (2003). Perennial computer camp "Tapkr". In: *International Conference "Information Technologies in Education"(ITO-2003)*, Moscow

KPFU (n.d.). *Contester system of the Kazan Federal University*. `http://acm.kpfu.ru`

topcoder (n.d.). *Contester System Topcoder*. `http://topcoder.com`

R. Hadiev is a senior teacher of Institute of Computational Mathematics and Information Technologies of Kazan Federal University. He has 30 years experience of work with talented students of university and schools. Coach of Kazan Federal University teams.



K. Khadiev is a lecture teacher of Institute of Computational Mathematics and Information Technologies of Kazan Federal University, researcher of Quantum Informatics laboratory of the Kazan Federal University, data scientist of Provectus Inc. He took part in All Russia school contests, ACM ICPC contests, GCJ, Huckercup and others. He is coach of Kazan Federal University teams. He has scientific papers in the computational complexity, automata theory, communication complexity and other areas.