Olympiads in Informatics, 2009, Vol. 3, 26–37

Tasks and Training the Intermediate Age Students for Informatics Competitions

Emil KELEVEDJIEV
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences
Akad. G. Bonchev str., block 8, 1113 Sofia, Bulgaria
e-mail: keleved@math.bas.bg

Zornitsa DZHENKOVA
Mathematical High School
2 Elin Pelin str., 5300 Gabrovo, Bulgaria
e-mail: zornica.dzhenkova@gmail.com

Abstract. The preparation of informatics competitions for an intermediate age group of school students that includes 14 and 15 years old children has become more important because of the possible introduction of new kinds of international competitions, like the recent establishment of the regional Balkan Youth Olympiad for students aged up to 15.5 years. Our paper presents the Bulgarian experience of training, where the main tools for teaching are tasks. The paper continues the previous work of the authors (Kelevedjiev and Dzhenkova, 2008a), where beginners group's tasks were arranged and classified according to some chosen set of keywords. The present study examines the tasks given at the Bulgarian national competitions for the intermediate age group during the last eight years. Some features of these tasks are discussed and compared to the tasks of the youngest age group.

Key words: tasks in competitive informatics, informatics for the intermediate age school students.

1. Introduction

In recent years (2001–2008), the Bulgarian national competitions in informatics for school students have developed as a system that includes three National Competitions (autumn, winter, and spring tournaments) and the National Olympiad with three rounds (Manev *et al.*, 2007). The students are divided into 5 age groups A, B, C, D, and E, which comprised 11–12, 9–10, 7–8, 6, and 4–5th school grades, respectively. In the Bulgarian schools, the mentioned grades correspond to 18–19, 16–17, 14–15, 13, and 11–12 years old children, respectively.

The classifications of tasks given at Bulgarian competitions for the age group E and D (youngest beginners) is presented in (Kelevedjiev and Dzhenkova, 2008a; Kelevedjiev and Dzhenkova, 2008b), where an attempt is made introducing key-words to indicate basic task features from 3 different points of view: (1) programming language elements; (2) control constructions; and (3) algorithms.

For the intermediate age group (group C) we chose as a classification principle one point of view only – which algorithmic approach is necessary to apply in order to solve the task.

2. Classification

Our research into the contents and solution methods for the tasks of the intermediate group given at the Bulgarian national competitions during the period 2001–2008 outlines that the following main algorithmic topics have been used:

a) *Counting and Combinatorics* including enumeration.

b) *Searching and Exhaustive search*. These topics include some improved searching methods as well as backtracking approaches.

c) *Geometry* with the predomination of problem on rectangular grids as well as tasks about sets of segments and rectangles with their sides parallel to coordinate's axes. Also presented are tasks involving plane and elementary geometry, and there are even tasks about solid geometry and tiling problems.

d) *Dynamic programming* with a typical example being the task about finding the longest common substring of two given strings.

e) *Graphs* with examples for tasks including finding connected components, shortest paths, matching problems, Euler cycles, cliques, etc.

f) *Digits* from a number, *arithmetic* and *number theory* including factorization, prime numbers, fractions, etc.

g) *Data structures* like stacks and queues.

h) *Others*, which refers to the tasks that are harder to classify. The reader may get some concepts of what such kinds of tasks look like by browsing the example tasks 3.8, 3.9 and 3.10, given below.

For some tasks, it is naturally to use two or more of the topics listed above for adequate classification. See also the table in the Appendix, where some topics are given together with subtopics.

3. Example Tasks

The following tasks are chosen to present several main topics and trends in the competitive informatics for the intermediate age group in the Bulgarian national competitions.

3.1. *Topic "Counting"* (*Task 2003, NOI-2, 2. Symmetric numbers*)

A given positive integer is call symmetric, if it is read in the same way from left to right and from right to left. For example: 474 and 2002. Write a program `Count` that on a given integers a, and b, computes how many symmetric integers exist in the interval $[a, b]$, $1 \leqslant a \leqslant b \leqslant 999999999$.

Example input: 9 23. Output: 3.

3.2. *Topic "Enumeration"* (*Task 2001, NOI-2, 1. Sequence*)

Consider the sequence: 1, 2, 3, ..., 9, 12, 13, ..., 19, 21, 23, ..., 98, 123, 124, ..., 987654321, where in an ascending order are arranged all positive integers that have no '0'-s in their decimal representation, and for which, all digits are distinct. Write a program **Num** that on given element x in the above sequence finds its position number k in the sequence, and vice versa, on given position number k, outputs the corresponding element in the sequence. The input file contains values of x and k. The output should contain the corresponding answers.

Example input: 21 11. Output: 18 13.

3.3. *Topic "Searching"* (*Task 2007, AT, 1. Numeric table*)

In each cell of a rectangular table of m rows and n columns, a positive integer is written. The largest value of these integers is less than 100. Three cells in the table are called neighbors, if every one of them has a common side with at least one of the others. Write a program `Maxi` that computes the largest sums which can be obtained by adding integers in three neighboring cells. On the first line in the standard input, values of m and n ($1 < m < 10$, $1 < n < 10$) are written. Each of the next m lines contains n integers to describe the corresponding table row.

Example input:

```
2 2
1 3
2 4
```

Output:

```
9
```

3.4. *Topic "Searching"* (*Task 2001, AT, 1. The best*)

Given are N ($5 \leqslant N \leqslant 20000$) positive integers, each less than 20 000, and a positive integer K ($3 \leqslant K \leqslant 100$, $K < N$). Write a program `Maxk` that finds the largest K integers among the given ones. The first two values in the input are N and K, followed by the given integers. The output should contain the largest found K integers in increasing order.

Example input: 8 4 133 74 12 38 29 56 19 31. Output: 38 56 74 133.

3.5. *Topic "Searching"* (*Task 2002, ST, 3. Interval*)

Given is a sequence of N integers $A_1, A_2, \ldots, A_N$ ($5 \leqslant N \leqslant 10000$, $-20000 \leqslant A_K \leqslant 20000$). We call an interval any subsequence of contiguous elements $A_L, A_{L+1}, \ldots, A_M$ ($1 \leqslant L \leqslant M \leqslant N$), such that for any A_K, $L \leqslant K \leqslant M$ it holds that the value of A_K is between the values of A_L and A_M. Write a program `Interval` that finds the longest interval in the given sequence. The input contains the value of N, followed by the integers of the given sequence. The output should contain two integers L and M that determine the longest interval.

Example input: 7 2 –3 –1 2 6 4 0. Output: 2 5.

3.6. *Topic "Geometry on grid"* (*Task 2001, AT, 3. Ice piece*)

A piece of ice is depicted as a set of cells in a table of N rows and N columns ($8 \leqslant N \leqslant 200$). In the table, all cells in the first and the last rows and columns are empty. The piece of ice is starting to melt. In an hour, the ice melts in the cells that have at lest two empty neighbors (above, below, left, or right). The ice in the other cells remains unchanged. On the right-hand part of Fig. 1 it is shown what occurs after an hour, starting from the configuration depicted on the left-hand part. Write a program `Ice` that computes how many hours are needed for all pieces of ice to melt. The input contains the value of N, followed by N rows, each contained N characters '0' or '*', where '0' denotes an empty cell, and '*' denotes a cell with ice. The output should contain the number of hours needed for the total melting of the ice.

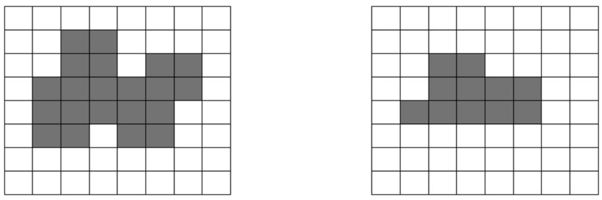


Fig. 1.

Example input:
8
00000000
00**0000
00**0**0
0******0
0*****00
0**0**00
00000000
00000000
Output:
4

3.7. *Topic "Dynamic programming"* (*Task 2001, AT, 2. Common substring*)

Given two strings A and B, each containing at least 3 and at most 250 decimal digits, we call the maximal common substring of A and B the longest string that is a substring in A and B.

Examples: For strings 112135349 and 66353, the maximal common substring is 353; the strings 112135349 and 67887 have no common substring. Write a program `Subs` that outputs the maximal common substring of two given strings.

3.8. *Topic "Others"* (*Task 2004, NOI-2, 2. Curious sequence*)

Consider the following integer sequence: 1, 11, 21, 1211, . . . , that starts with 1 and every element describes the previous one: as an example the third element is 21, because the forth element tells us one '2' and one '1' . Write a program `Selfdes` that on given input integers n and k ($0 < n < 10^9$, $0 < k < 20$) outputs the k-th element of the above sequence, where the first element of the sequence is n.

Example input: 2 4. Output: 3112.

3.9. *Topic "Others"* (*Task 2006, ST, 2. Matrix*)

Given is a matrix containing integers (between 1 and 10 000) with N rows and M columns ($2 \leqslant N \leqslant 1000$, $2 \leqslant M \leqslant 1000$). Two rows are called similar, if the first of them can be obtained by rearranging the elements of the second row. Write a program `Matrix` that outputs the maximal size of a set of matrix's rows such that no any two of them are similar. The input contains the values of N and M, followed by N lines, each containing M integers separated by spaces to describe the Nth row of the given matrix.

Example input:

5 4
10 1000 5 200
70 110 70 30
5 200 10 1000
4 6 11 45
70 70 30 110

Output:

3

3.10. *Topic "Others"* (*Task 2005, NOI-2, 2. Derivatives*)

For a given positive integer p, we denote by p' another positive integer, which is called first order derivative of p, so that the following rules hold:

1) $1' = 0$;
2) $p' = 1$, if p is prime;
3) $p' = (ab)' = a'b + ab'$, where a and b are divisors of p.

Applying the same rule to a derivative of the first order, we obtain the corresponding derivative of the second order, and analogously, we are able to compute a derivative of kth order using the $k - 1$ order. Write a program `Derive` that on given positive integers p and k ($p \leqslant 1000$, $k \leqslant 10$) computes the kth order derivative of p.

Example input: 16 3. Output: 176.

4. Trends

In Table 1, we present cumulative data about the number of tasks with assigned main algorithmic topics. The study is based on the text descriptions of the tasks (taken from (Bulgarian web portal site for competitions, 2009; Bulgarian web site for school competitions, 2009)) for the intermediate age group given at the Bulgarian national competitions in informatics during the period 2001–2008. The reader may refer to the Table 5 in the Appendix for the list of tasks.

We display diagrams to illustrate observed tendencies for monotonic or periodic trends in time appearance for the number of tasks from specific types (by means of algorithm topic involved) during the period 2001–2008 in the scene of the Bulgarian national competitions in informatics for the intermediate age group (Figs. 2–4).

As a measure of difficulty for a particular task we adopt the following 3 values:

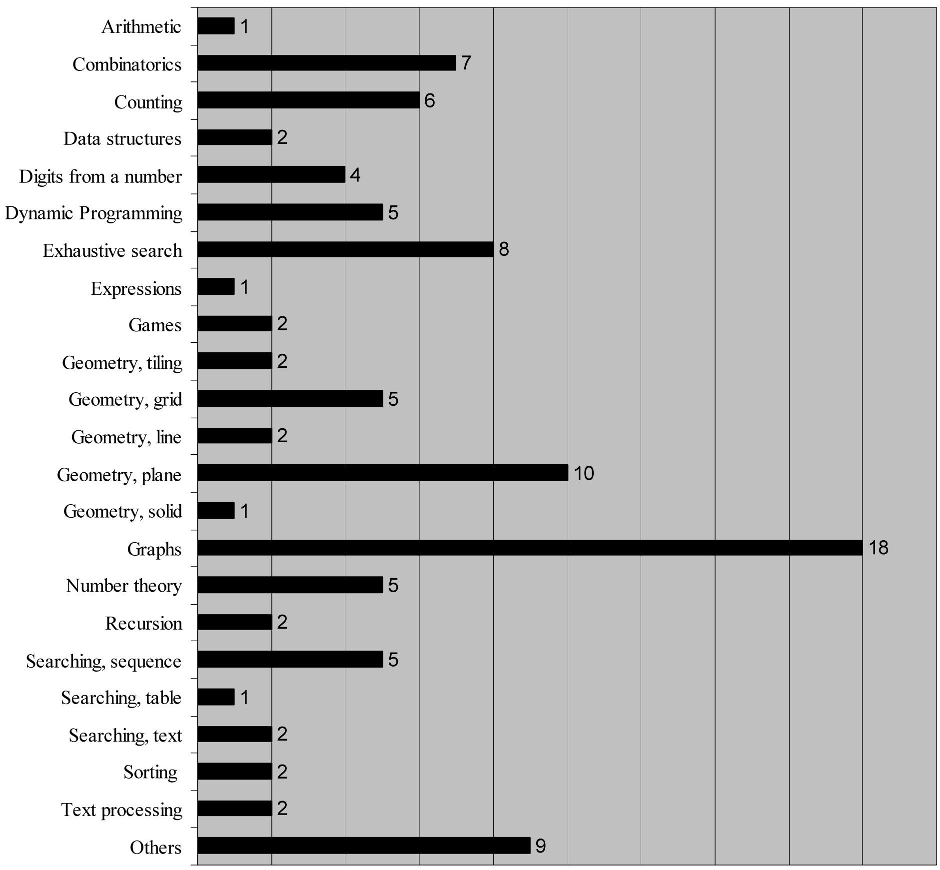
a) the percentage of students who solve the task by gaining the maximal score;

b) the percentage of students who do not solve the task at all;

c) the ratio of the above two values.

Table 1

Number of tasks with assigned main algorithmic topics



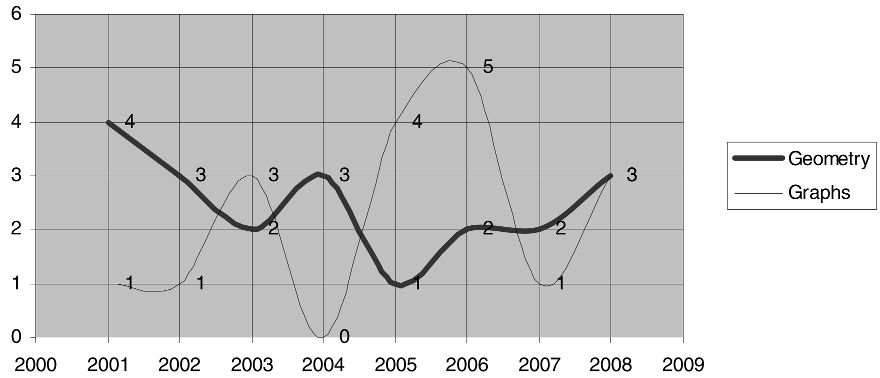


Fig. 2. Number of tasks belonging to the groups “Geometry” and “Graphs” during the years.

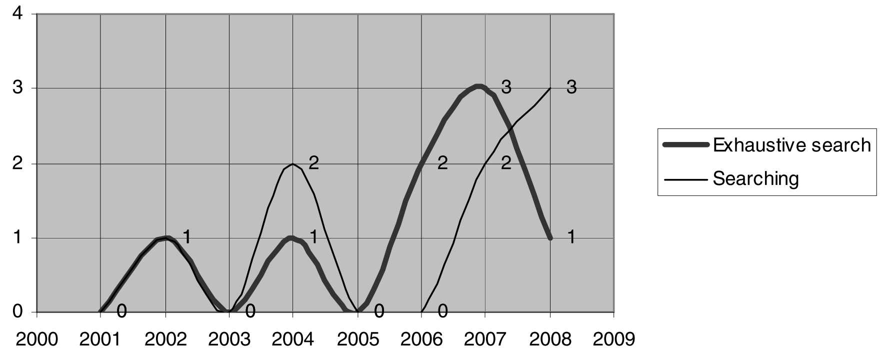


Fig. 3. Number of tasks belonging to the groups “Exhaustive search” and “Searching” during the years.

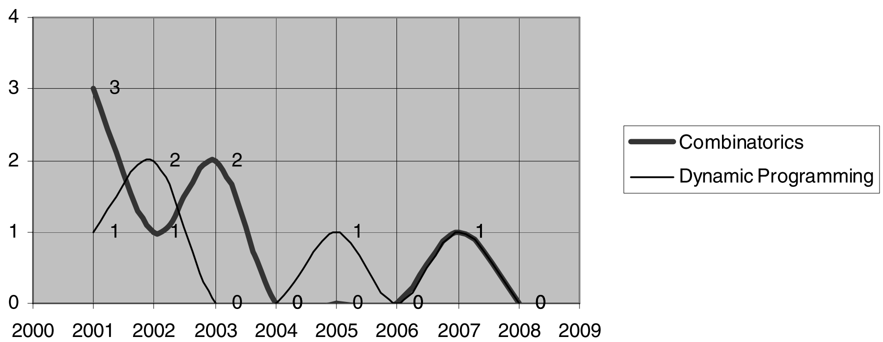


Fig. 4. Number of tasks belonging to the groups “Combinatorics” and “Dynamic programming” during the years.

Computing these values for the results of the tasks groups using the disposable data, we present Tables 2, 3 and 4.

5. Conclusions

Although the data presented in the above tables and graph samples are not statistically significant, they give us some ideas about the variety of themes. Assigning topics to each task is influenced by personal opinions, but there are some more or less steady principles for choosing them. In many cases the topic names are self-descriptive and publishing information about tasks together with these keywords is easily understandable and can help teachers in their training process with students, as well as help the authors of tasks for future competitions.

Comparing with the similar study of task classification for the beginner's group (Kelevedjiev and Dzhenkova, 2008a; Kelevedjiev and Dzhenkova, 2008b) we can observe here the underlying role of algorithms which is influenced by an increase of tasks difficulty for the intermediate group.

The value called "ratio" in the Tables 2, 3, and 4, gives a good estimation in our opinion for both measures: the difficulty of the tasks for students and the appropriateness of choosing the tasks by the organizers of the competition. As an example, the tasks from the "special" group, called "others", have the minimal value for "ratio".

Table 2

Geometry tasks: excellence and poor results

Task name	**Year**	**% excellence**	**% poor**	**ratio**
ice piece	2001	12.50	62.50	0.20
tiles	2002	0.00	32.56	0.00
dominoes	2003	2.17	60.87	0.04
movement	2003	39.29	32.14	1.22
kingdoms	2004	3.85	76.92	0.05
rectangles	2004	0.00	50.00	0.00
abc	2004	2.38	69.05	0.03
ruler	2005	40.54	40.54	1.00
rectangles	2006	12.35	33.33	0.37
darts	2006	0.00	87.50	0.00
jumps	2007	21.43	17.86	1.20
crossing	2007	3.51	91.23	0.04
segments	2008	29.03	41.94	0.69
move	2008	15.00	45.00	0.33
angles	2008	13.89	41.67	0.33
Average		**13.06**	**52.21**	**0.37**

Table 3

Graphs tasks: excellence and poor results

Task name	**Year**	**% excellence**	**% poor**	**ratio**
path	2002	4.35	52.17	0.08
teleporting	2003	4.00	84.00	0.05
expedition	2003	0.00	84.00	0.00
islands	2003	28.57	42.86	0.67
tree	2005	26.98	47.62	0.57
gnomes	2005	6.25	34.38	0.18
tour	2005	0.00	80.77	0.00
school	2005	23.33	53.33	0.44
marriage	2006	1.79	73.21	0.02
phones	2006	10.00	66.67	0.15
trade	2006	53.33	36.67	1.45
plate	2006	0.00	51.28	0.00
mate	2006	20.00	43.33	0.46
man	2007	7.14	51.79	0.14
pebbles	2008	10.53	47.37	0.22
friends	2008	11.11	44.44	0.25
triangles	2008	36.36	18.18	2.00
Average		**14.34**	**53.65**	**0.39**

Table 4

Excellence and poor results for groups of tasks

Tasks' type	**% excellence**	**% poor**	**ratio**
Searching	18.89	44.45	1.35
Dynamic programming	12.84	59.83	0.43
Graph	14.34	53.65	0.39
Geometry	13.06	52.21	0.37
Counting	5.51	42.06	0.25
Others	6.27	50.15	0.20
Total Average	**11.77**	**52.59**	**0.39**

Appendix

Table 5 presents all tasks given at the Bulgarian competitions for the intermediate age group during the years 2001–2008. In the column "Competition", the names of the autumn, winter and spring tournaments are abbreviated as AT, WT and ST, respectively, and the second and third rounds of the Bulgarian National Olympiads in Informatics are denoted by NOI-2, and NOI-3, respectively.

Table 5

Tasks given at the Bulgarian competitions for the intermediate age group during the years 2001–2008

No	Year	Competition	Problem No	Problem Name	Method
1	2001	AT	1	best	Searching, Sorting
2	2001	AT	2	common substring	Dynamic Programming
3	2001	AT	3	ice piece	Geometry, Grid
4	2001	NOI-2	1	num	Combinatorics, Enumerating
5	2001	NOI-2	2	line	Geometry, Plane
6	2001	NOI-2	3	maze	Graphs, Cycles
7	2001	ST	1	three	Geometry, Plane
8	2001	ST	2	war	Games
9	2001	WT	1	table	Combinatorics
10	2001	WT	2	ab	Digits
11	2001	WT	3	line	Geometry, Plane
12	2001	WT	4	sum	Combinatorics
13	2002	AT	1	path	Graphs
14	2002	AT	2	frequencies	Counting, Text
15	2002	AT	3	digit	Searching, Sequence
16	2002	NOI-2	1	table	Counting, Tables
17	2002	NOI-2	2	tiles	Geometry, Plane
18	2002	NOI-2	3	concert	Dynamic Programming
19	2002	ST	1	molecules	Combinatorics
20	2002	ST	2	bed	Dynamic Programming
21	2002	ST	3	Interval	Exhaustive search
22	2002	WT	1	primes	Number theory
23	2002	WT	2	para	Geometry, Solid
24	2002	WT	3	path	Geometry, Grids
25	2003	AT	1	lighting	Combinatorics
26	2003	AT	2	movement	Geometry, Grids
27	2003	AT	3	islands	Graphs, Grids
28	2003	NOI-2	1	Egypt fractions	Number theory
29	2003	NOI-2	2	count	Counting, Enumeration
30	2003	NOI-2	3	dominoes	Geometry, Tiling
31	2003	ST	1	expedition	Graphs, Matching
32	2003	ST	2	expressions	Expressions
33	2003	ST	3	teleporting	Graphs, Shortest path
34	2003	WT	1	shuffle	Combinatorics
35	2003	WT	2	tail	Data structures, queue
36	2003	WT	3	young	Sorting
37	2004	AT	1	supermarket	Data structures, queue
38	2004	AT	2	code	Searching, Text
39	2004	AT	3	temperature	Searching, Sequence
40	2004	NOI-2	1	enemy	Arithmetic, Fractions
41	2004	NOI-2	2	self-description	Others, Recursion
42	2004	NOI-2	3	abc	Geometry, Plane
43	2004	ST	1	grades	Others
44	2004	ST	2	squares	Exhaustive search
45	2004	ST	3	primes	Number theory
46	2004	WT	1	barcode	Others, strings
47	2004	WT	2	rectangles	Geometry, Plane
48	2004	WT	3	kingdoms	Geometry, Grid
49	2005	AT	1	hyperlinks	Text processing

To be continued

Table 5 (continued)

No	Year	Competition	Problem No	Problem Name	Method
50	2005	AT	2	school	Graphs, Connected components
51	2005	AT	3	numbers	Counting
52	2005	NOI-2	1	tree	Graphs
53	2005	NOI-2	2	derivative	Others, Computations
54	2005	NOI-2	3	simple	Counting
55	2005	ST	1	gnomes	Graphs, Paths
56	2005	ST	2	two	Games
57	2005	ST	3	row	Others, Sequences
58	2005	WT	1	ruler	Geometry, Line
59	2005	WT	2	run	Dynamic Programming
60	2005	WT	3	tour	Graphs, Cycles
61	2006	AT	1	right primes	Number theory
62	2006	AT	2	plate	Graphs, Euler cycles
63	2006	AT	3	darts	Geometry, Plane
64	2006	NOI-2	1	rectangles	Geometry, Plane
65	2006	NOI-2	2	bmax	Digits
66	2006	NOI-2	3	delmin	Number theory
67	2006	NOI-3	1	mate	Graphs
68	2006	NOI-3	2	trade	Graphs, Paths
69	2006	NOI-3	3	min	Others, Tables
70	2006	ST	1	socks	Others, Tables
71	2006	ST	2	matrix	Exhaustive search
72	2006	ST	3	marriage	Graphs, Matching
73	2006	WT	1	Morse code	Others, Strings
74	2006	WT	2	Puzzle	Exhaustive search
75	2006	WT	3	Phones	Graphs, Paths
76	2007	AT	1	numerical table	Searching, Table
77	2007	AT	2	puzzle	Exhaustive search
78	2007	AT	3	names	Counting
79	2007	NOI-2	1	similar	Others, Tables
80	2007	NOI-2	2	minimal	Exhaustive search
81	2007	NOI-2	3	inequality	Exhaustive search, Arithmetic
82	2007	ST	1	man	Graphs, Connected components
83	2007	ST	2	segments	Dynamic Programming
84	2007	ST	3	jumps	Geometry, Line
85	2007	WT	1	sequence	Searching, Sequence
86	2007	WT	2	crossing	Geometry, Plane
87	2007	WT	3	necklace	Digits
88	2008	AT	1	mirror	Searching, Sequence
89	2008	AT	2	pebbles	Graphs, Games
90	2008	AT	3	bank	Combinatorics, Coding
91	2008	NOI-2	1	friends	Graphs, Connected components
92	2008	NOI-2	2	angles	Geometry, Plane
93	2008	NOI-2	3	justify	Others, Text processing
94	2008	NOI-3	1	move	Geometry, Grid
95	2008	NOI-3	2	trans	Digits
96	2008	NOI-3	3	sword	Exhaustive search, Strings
97	2008	ST	1	disk	Others, Recursion
98	2008	ST	2	segments	Geometry, Tiling
99	2008	ST	3	subsequences	Searching, Sequence
100	2008	WT	1	future	Others, Long integers
101	2008	WT	2	triangles	Graphs, Clique
102	2008	WT	3	context	Searching, Text

References

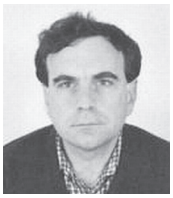
Manev, K., Kelevedjiev, E. and Kapralov, S. (2007). Programming contests for school students in Bulgaria. *Olympiads in Informatics International Journal*, **1**, 112–123.

Kelevedjiev, E. and Dzhenkova, Z. (2008a). Tasks and training the youngest beginners for informatics competitions. *Olympiads in Informatics International Journal*, **2**, 75–89.

Kelevedjiev, E. and Dzhenkova, Z. (2008b). Competition's tasks for the youngest school students. In *Mathematics, Informatics and Education in Mathematics and Informatics*, *Spring Conference of the UBM*, Borovetz.

Bulgarian Web Portal Site for Competitions in Informatics. Retrieved 1 March 2009 from `http://infoman.musala.com`

Bulgarian Web Site for School Competitions in Informatics. Retrieved 1 March 2009 from `http://www.math.bas.bg/infos`



E. Kelevedjiev is a research fellow in the Institute of Mathematics and Informatics at the Bulgarian Academy of Sciences. His field of interests includes algorithms in computer science, operation research, digitization techniques, etc. He is a chairman of the Bulgarian National Committee for Olympiads in Informatics; leader or deputy leader of the Bulgarian teams for many IOI's and BOI's.

Z. Dzhenkova is a teacher in the Mathematical High School in Gabrovo, Bulgaria. She is coauthor of a manual for beginner's training in competitions and olympiads in informatics. Her field of scientific interests includes education in informatics and information technology; leader of school student teams and instructor in competitive informatics.