Olympiads in Informatics, 2013, Vol. 7, 101–112
© 2013 *Vilnius University*

Tasks in Informatics of Continuous Content

Pavel S. PANKOV
International University of Kyrgyzstan
e-mail: pps50@rambler.ru

Abstract. Almost all tasks at informatics olympaids are of discrete content. Tasks of continuous content are rare; moreover, some of them are not algorithmic in nature or it is not possible to score their solutions strictly because of using approximate calculations. We propose to involve such tasks with strict formulations and discrete (in integer numbers) solutions by means of ideas of interval analysis and present some ways to create and to solve them. We hope that some classes of such tasks would enlarge scope of tasks for use in informatics olympiads at various levels.

Key words: olympiads in informatics, tasks, continuous content, validating computations, interval analysis.

1. Definitions

First of all, note that all tasks of types proposed below can be written without special terminology, for example:

Task 1.1. Given an integer number $N,\ 10 < N < 10^{1000}$. Find an integer K that the solution of the equation $x^3 + x = N$ fulfills the assertion $K \leqslant x < K + 1$.

The knowledge of the formula for solution of cubic equation is a disadvantage in solving of this task because it is impractical while the common bisectional search method yields result easily.

The following considerations are developing of idea of this task:

Denote the space of real numbers as R. We will consider continuous objects, such as real numbers (basic objects), finite ordered sets of real numbers (vectors in R^n), polygons, polynomials, and continuous functions defined in any way.

To make such definitions strict, they will be presented (in the task conditions) by means of integer numbers.

It is known that not all $x \in R$ can be represented by any algorithm (by constructive methods). Hence, the notion of computable number was introduced (see, for instance, Shanin, 1968):

DEFINITION 1.1. Any $x \in R$ is said to be computable if there exists an algorithm which reworks any natural number n in such integer m that $|x - \frac{m}{n}| < \frac{1}{n}$.

Certainly, such definitions will not be used in tasks but properties of computable numbers ought to be kept in mind to produce correct statements of problems.

In our opinion, computability of real numbers arising in tasks is obvious for the common contestant.

By strict solution we mean the following

DEFINITION 1.2 (Pankov *et al.*, 1979). Computations are said to be validating if they are conducted in such a way that their results being interpreted as sets of continuous objects contain the true objects (solutions of proposed tasks).

The term "reliable computations" is also used.

One of ways to implement validating computations is interval analysis (Moore, 1967). Let $X = [x_-, x_+] \subset R$. If $x \in X$, then X is said to be an interval presentation of x; if x_- and x_+ are rational numbers and are presentable in a computer in any standard way then X is said to be a machine interval presentation of x.

For an interval $X = [x_-, x_+\ ]$, the difference $Wid(X) = x_+ - x_-$ is said to be the width of X; let the width of an interval vector be the maximum of widths of its components.

The narrower X, the better the presentation.

If the width of an interval (vector) is zero then it is said to be degenerate, i.e., it is equivalent to a single number (vector).

For intervals, the notions *Length*(X) and *Wid*(X) coincide. For interval vectors the notions *Measure* (*Area* for 2D, *Volume* for 3D ...) differ from *Wid*.

We will use the functions of an interval considered as an entire object: $X_- := x_-, X_+ := x_+$; if X is integer and $Wid(X) > 1$ then *midX* $:= int((X_- + X_+)/2)$ (for instance, *int* means rounding down to an integer);

splitting: *Lhalf* $(X) := [X_-,$ *mid*$(X)]$ and *Rhalf* $(X) := [$*mid*$(X), X_+\ ]$.

The last two operations are also applicable to interval vectors (for instance, splitting by the widest component; if there are some such ones then choose one on the least position).

Also, we will use the function of outer interval presentation of finite or other bounded sets in R^n: $Outer(W)$ is the "least" (the narrowest in all components) box (interval for $n = 1$, interval vector for $n > 1$) containing the set W. The simplest example: $Outer(\{a, b\}) = [\min\{a, b\}, \max\{a, b\}]$.

(We will use operations *min* and *max* both for finite and bounded closed infinite sets of R).

Recall the obvious formulas of interval analysis:

$$\begin{aligned}&[x_-, x_+] + [y_-, y_+] = [x_- + y_-, x_+ + y_+]; [x_-, x_+] - [y_-, y_+]\\ &\quad = [x_- - y_+, x_+ - y_-],\\ &[x_-, x_+] \cdot [y_-, y_+] = Outer(\{x_-y_-, x_+y_+, x_-y_+, x_+y_-\})\end{aligned}$$

(by means of branch on condition the last formula can be reduced to two products); if (n *is odd or* $0 \notin [x_-, x_+\ ]$) then $[x_-, x_+]^n = Outer(\{x_-^n, x_+^n\})$ else $[x_-, x_+]^n = Outer(\{x_-^n, 0, x_+^n\})$.

Interval analysis uses the triple logic. Let $x \in X \subset R$ and we try to prove that $x > 0$. If $x_+ < 0$ (*in other words*, $X < 0$) then $x < 0$ (result: *No*); if $x_- > 0$ (*in other words*,

$X > 0$) then $x > 0$ (result: *Yes*); elsewhere ($0 \in X$; result: *Uncertainty*; try to calculate a narrower interval presentation for x).

The main idea of interval analysis is deriving results on infinite sets by means of finite operations on boundaries of intervals.

Remark 1.1. Strict results of approximate calculations could be also obtained by means of strict estimation of rounding error; but implementation of such estimation during actual vast computations is impossible.

Remark 1.2. Results of validating computations (of interval analysis) can be interpreted in two main ways: as a result for a single number or as result for a set of numbers. Give a simplest

EXAMPLE 1.1. Suppose that the statement "$\pi \in [3.1, 3.2]$" is proven already. Due to rules of interval analysis, the calculation $[3.1, 3.2]^2$ yields the result $[9.6, 10.3]$ (directed rounding off to one decimal place). Thus, two statements are proven:
"$\pi^2 \in [9.6, 10.3]$"; "For all $x \in [3.1, 3.2]$, $x^2 \in [9.6, 10.3]$".

Recall definitions of interval analysis for functions.

If the inclusion $x_- \leqslant \text{ x} \leqslant x_+$ implies the inequality $p(x) \geqslant P_-(x_-, x_+)(p(x) \leqslant P_+(x_-, x_+)$ respectively) then a function $P_-(x_-, x_+)$ of two variables is said to be a minorant ($P_+(x_-, x_+)$ is said to be a majorant respectively) of the function $p(x)$ of one variable.

The interval function $P(x_-, x_+) = [P_-(x_-, x_+), P_+(x_-, x_+)]$ is said to be an interval representation of $p(x)$. If two interval representations fulfill the assertion $P_1(x_-, x_+) \subseteq P_2(x_-, x_+)$ then $P_1(x_-, x_+)$ is said to be not worse than $P_2(x_-, x_+)$.

Remark 1.3. $P * (x_-, x_+) = \mathit{Outer}(\{p(x) | x_- \leqslant x \leqslant x_+\})$ is the best interval representation for $p(x)$. But its exact calculation is practically impossible for functions arising in tasks. And the aim of interval analysis is constructing "sufficiently good" interval representations.

Interval analysis is used as follows: repetition of uniform computations by means of computer for any finite covering of a set yields a (strict) result for all points of this set. To diminish the number of elements of the covering, bisection method can be used.

We call intervals (interval vectors) with integer boundaries integer intervals (integer interval vectors correspondingly).

2. Correct Statement of Problems and Scoring

There was proven the following (see, for instance, Shanin, 1968)

Theorem 2.1. The problem of distinguishing the cases "$x < 0$", "$x = 0$" and "$x > 0$" for a computable number x is irresolvable algorithmically in general case (cf. "triple logic" above).

The narrowest non-degenerate integer interval has the width 1.

If given functions are computed exactly for integer numbers then the following statement of problem may be correct:

Statement 2.1. Find a semiopen integer interval of width 1 containing the solution, i.e., to find such integer K that $K \leqslant x < K + 1$.

But the general task to obtain such integer interval is incorrect: if the unknown number x is integer then methods of interval analysis in virtue of Theorem 2.1 can yield only a result of type $[x-\varepsilon, x+\varepsilon]$ where ε is a small positive number, and the narrowest attainable integer interval presentation is $[x - 1, x + 1]$.

So, the following two statements of problem are correct:

Statement 2.2. Find an integer interval (vector) containing the solution (real number or vector of real numbers respectively) of width not greater than 2;

Statement 2.3. Find an integer interval (vector) of width not greater than a given natural number (greater than 2).

There are two ways to score an answer given by the contestant (output-only task) or by the contestant's program:

- the jury knows the narrowest integer interval and an answer is to contain it;
- the scoring program checks an answer by a posteriori computations.

Remark 2.1. Sometimes, to substantiate an answer, using of some mathematical results (theorems) and/or some additional calculations (checking-up of some conditions) is necessary. That is why proposals to look through listings of contestants' programs arise time-by-time. But the resolutions of majorities of juries are same: such looking through of all listings is practically impossible. So, the corresponding theorems of analysis will be mentioned below but using of them, certainly, is not to be checked in listings.

Moreover, for types of tasks proposed below the contestants can improvise, invent their own algorithms including heuristic ones, use a posteriori methods, use real numbers (with rounding-off of final results) etc. We consider this as a positive effect making competitions more interesting.

Certainly, the jury is to prepare tests with strict methods:

- interval computations mentioned in this paper;
- fitting of functions.

Simplest examples of fitting: choose a polynomial $p_1(x) > 0 (x \in R)$. If $p_2(x) = a + (x - b)^2 p_1(x)$ (*after removing brackets*) then $\min\{p_2(x)\} = a$; if $p_3(x) = (x - b)p_1(x)$ (*after removing brackets*) then the solution of the equation $p_3(x) = 0$ is b.

3. Main Types of Tasks and Mathematical Methods to Solve Them

Below appropriate one of Statements 2.1, 2.2 or 2.3 is meant in contents of all tasks (under the general denotation Statement 2 with the upper boundary w for width).

General task 3.1. Given a function in a domain $G \subset R^n$

$$f : G \to R;$$

find an integer interval containing its least (greatest) value (and an integer interval containing the value of argument yielding all solutions correspondingly).

General task 3.2. Given an equation in a domain of type

$$f(x) = 0 \quad (x \in G),$$

find an integer interval (all integer intervals) containing its least (greatest) solution (all solutions correspondingly).

General task 3.3. Given a geometrical object, find an integer interval containing any measure (length, area, volume) of it.

Remark 3.1. Experience of using such tasks at olympiads gave the following paradoxical fact. Contestants who knew "formulas" for the values to be calculated demonstrated worse results than those who did not study or have forgotten such "formulas".

The explanation of this fact is following. So-called "formulas", "expressions in explicit form" are not irrevocable results (although they seem to be such ones) but they are connections between some mathematical notions only.

Therefore, we propose to teach mathematics to computer scientists with stressing the following statements:

– every mathematical method is to be considered from the viewpoint of its constructiveness;
– common (approximate) calculations do not yield guaranteed results;
– strict estimation of computational error in real tasks is impossible (what would be also useful for their forthcoming professional activity).

Such a subject was implemented by us (Pankov *et al.*, 1996, 2002). Exposition of each notion begins with conditions ensuring its computability; further, formulas and ways of better computations are given. For instance, "differentiation" is not a constructive operation (and many computer scientists lost and lose a lot of time because of this fact) while "integration" is a "good" operation. Finding values of maximum and minimum are constructive operations under wide assumptions but finding points (arguments) where these values are attained is not constructive. Also, mistakes in programming being very difficult to be found arise because of ignoring Theorem 2.1 above.

Moreover, we (Pankov, 1987) declared that "an expression in explicit form" is not a mathematical notion but a historical one. Namely, if any type of equations arose in mathematical investigations frequently then their solutions are given special denotations; if these denotations are apt then their properties are investigated thoroughly, tables of and (last fifty years) computer procedures to calculate their values approximately are developed.

To warrant this conclusion, list

constants : $\tau = \frac{\sqrt{5}+1}{2}$ (*golden ratio*), π, *e, Euler constant C*, ...

functions: *roots, trigonometric functions, inverse trigonometric functions,* $\exp$, $\log$ and being infinitely replenished list of so-called *special functions:* integrals which cannot be expressed by means of preceding functions: *integral sine, integral logarithm*, . . ., *Euler B- and Γ- functions*, *Bessel functions, hypergeometric function, Riemann ζ- function*, . . .

Also, recall a standard expression in text-books and papers: *Given a polynomial. Let* $x_1, x_2, \ldots$ *be its roots* . . . But the task of obtaining roots by given coefficients is separate and complicated.

Further, mathematicians try to reduce solving of other types of equations to those. But existence of a denotation is not a solution yet. So, we proposed not "to express a continuous object in explicit form" but to try to prove computablility of continuous objects.

4. Validating Assertions for Polynomials

We will consider tasks for polynomials with integer coefficients only. Square roots can also be involved. Combinations of polynomials, operations *max, min, abs*, square roots can yield indefinitely many tasks. For example, the function

$$f(x) = |c_1x^2 - \min\{c_2x + c_3, |c_4x + c_5x^3|\}| + c_6|x|$$

with given integer constants is sufficiently complicated to avoid all "analytical" methods.

All given numbers are assumed to be integer.

We will use mathematical denotations close to ones of algorithmic languages.

Given a natural number N and numbers $A[0], A[1], \ldots, A[N]$; let $A[N] > 0$ below.

Define a polynomial

$$p(x) := \Sigma\{A[n]x^n|n = 0..N\}. \tag{4.1}$$

The simplest interval representation of (4.1) is obvious:

$$P(x_-, x_+) := \Sigma\{A[n] \cdot [x_-, x_+]^n|n = 0..N\}. \tag{4.2}$$

There is the law of subdistributivity for intervals: $X \cdot (Y + Z) \subseteq \mathrm{X} \cdot Y + X \cdot Z$. Hence, a better interval representation (also, either minorant or majorant) can be constructed by means of presenting (4.1) by Horner method

$$p_N(x) := A[N]x; \textit{for } n = N - 1 \textit{ downto } 0 \ \{p_n(x) := x(p_{n+1}(x) + A[n])\};$$
$$p(x) \equiv p_0(x). \tag{4.3}$$

Consider also a function $q(x) = \frac{K}{p(x)}$ where K is a given natural number and it is given or proven that $p(x) \neq 0$; suppose that $p(x) > 0$ within a domain.

Then an interval representation of $q(x)$ is given by the formula

$$Q(x_-, x_+) := [\textit{floor}(K/P_+(x_-, x_+)),\ \textit{ceiling}(K/P_-(x_-, x_+))]. \tag{4.4}$$

If it is too rough then choose a natural number K_1 and define

$$Q * (x_-, x_+) := [\textit{floor}(K_1K/P_+(x_-, x_+)), \tag{4.5}$$

ceiling $(K_1K/P_-(x_-, x_+))$];

$Q(x_-, x_+) = \frac{Q*(x_-, x_+)}{K_1}$; the factor $\frac{1}{K_1}$ is being kept mentally for forthcoming computations.

5. Examples of Tasks

Tasks will be given in two versions: on bounded domain and on unbounded one. In the last case the domain by means of some mathematical assertions must be reduced to a bounded one.

Also, we will consider only one-dimensional case for given functions and two-dimensional case for given figures. Generalization to multi-dimensional cases is reasonable only for simple data because it itself is very complicated.

Task 5.1. Given $N > 3$, a polynomial $p(x)$ of degree N and two numbers $a < b$, find an integer interval containing $p_{\min} := \min\{p(x)|a \leqslant x \leqslant b\}$.

EXAMPLE: $\min\{6x^2 - 30x + 40| - 1 \leqslant x \leqslant 8\} \in [2, 3]$ ($N = 2$ here).

Remark 5.1. Such example is necessary; otherwise many contestants will take integer values of x only: $\min\{6x^2 - 30x + 40|x = -1..8\} = 4$.

Solutions for (4.1).

1st step. Choose and build any minorant $P_-(x_-, x_+)$ for $p(x)$.

2nd step. Choose a validating algorithm of global search $\min\{p(x);\ P_-(x_-,\ x_+)|x \in [a, b]\}$.

The simplest one is of exhaustive search:

Algorithm 5.1. *Calculate* $P_{\min} = [\min\{P_-\ (k, k+1)|k = a..b-1\},\ \min\{p(k)|k = a..b\}]$.

If it does not meet Statement 2 then try to build a better minorant and repeat calculations.

More effective algorithms use bisectional search.

Describe one of them demanding the least volume of memory: successful proving of inequalities (lower bounds for $p_{\min}$).

Algorithm 5.2.

```
Denote A:= $[a, b]$;
Define array $X[]$ of intervals;
$p_{--} := P_-(A);\ p_{-+} := p(midA)$;
$X[1] := Lhalf(A); X[2] := Rhalf(A)$;
$K := 2$;
repeat
{$p_m :=$ mid([$p_{--},\ p_{-+}$ ]);
repeat
{$if P_-(X[K]) \geqslant p_m$
*then* K:= $K-1$
*else*
$\{p_{-+} := \min\{p_{-+},\ p(mid(X[K])\};\ p_m := textitmid[p_m, p_{-+}];$
$X[K+1] := Lhalf(X[K]);\ X[K] := Rhalf(X[K]);$
$K := K+1\}$
$\}$
*until* K= $0$;
$p_{--} := p_m\}$;
*until* ( $Wid(X[K]) = 1$ *or* $Wid([p_{--}, p_{-+}]) \leqslant w$ *// Statement 2//*);
```

Result. $p_{\min} \in [p_{--}, p_{-+}]$.

If the algorithm stopped due to restriction on time then a better minorant is necessary; if it done due the occurrence $Wid(X[K]) = 1$ then

– also a better minorant is to be involved;

– if we cannot do it then we are to involve narrower (fractional) intervals.

The most standard way is following. Choose a natural K_1 and substitute $x = \frac{x_1}{K_1}$, x_1 is integer, into $(4.1) : p(x) = K_1^{-N}\Sigma\{A[n]K_1^{N-n}x_1^n|n = 0..N\}$;

the factor K_1^{-N} is being kept mentally for forthcoming computations.

Such cross-partition is to be applied to intervals distinguished by Algorithm 5.2.

Remark 5.2 (cf. Remark 3.1). Those who know "formulas" would solve the task as follows. "Define $p'(x) = \Sigma\{A[n]nx^{n-1}|n = 1..N\}$, try to solve the equation $p'(x) = 0$. Let $x_1, x_2, \ldots, x_M (M \leqslant \text{ N} - 1)$ be its roots.

Calculate $p_{\min} := \min\{\min\{p(x_m)|(m = 1..M) \wedge (a \leqslant x_m \leqslant b\}, p(a), p(b)\}.''$

But this way for $N > 3 (N - 1 > 2)$ is much more complicated and contains practically irresolvable components.

Task 5.2. If N is even and greater than 2, find an integer interval containing
$p_{\min} := \min\{p(x)|x \in R\}$.

Solution for (4.1).

1st step. Find a priori boundaries for $\arg\min p(x)$. A rough estimation is derived from the assertion: if $p(x) > p(0)$ then $p(x) > p_{\min}$. Let $|x| \geqslant 1$:

$$\begin{aligned} p(x) - p(0) &\geqslant A[N]x^N - \Sigma\{|A[n]| \cdot |x|^n|n = 1..N-1\} \\ &\geqslant |x|^{N-1}(A[N]|x| - \Sigma\{|A[n]||n = 1..N-1\}). \end{aligned}$$

Thus, we obtain the following domain for search:

$|x| \leqslant x_0$:=*max{1, ceiling* $(\Sigma\{|A[n]||n = 1..N-1\}/A[N])\}$; and the task is reduced to Task 5.1.

Consider Task 3.2. There exists the algorithm finding the number of all (real) roots of a polynomial with integer coefficients but it is too complicated. Thus, additional conditions are to be put. The simplest (and correct) version is

Task 5.3. If $A[0] < 0; A[n] \geqslant 0 (n = 1..N)$, find an integer interval containing the (unique) solution of the equation $p(x) = 0, x > 0$.

To avoid exhaustive search, $|A[0]|$ is to be very large, for example, $A[0] < -10^{40}$ for $N = 4$.

Thus, a kind of long arithmetic is to be constructed.

Solution.

```
1st step.
$x_0 = 1$;
repeat { $x_0 := 2x_0$ } until $p(x_0) > 0$.
```

2nd step. Use bisectional search (with rounding-off to integer, cf. the operation *mid*) on $[0, x_0]$ (or on $[x_0/2, x_0]$).

Task 5.4. Given $N > 2$, (large) natural K, a polynomial $p(x)(A[0] > 0, A[n] \geqslant 0 | n = 1..N)$ and two positive numbers $a < b$, find an integer interval containing the area s between the x-axis, the graph of the function $q(x) = \frac{K}{p(x)}$ and lines $x = a$ and $x = b$.

Remark 5.3. We evade the term "integral", cf. Remark 3.1.

Solution. Construct an interval representation for the function $q(x)$, see (4.4) and (4.5). The exhaustive summation for (4.5):

$$s \in \Sigma\{Q * (x, x+1) | x = a..b-1\}/K_1.$$

If the number $(b - a)$ is too large then bisectional partition can be used.

Algorithm 5.3.

```
Define arrays $X[], V[]$ of intervals.
$X[1] := [a, b]$;
$V[1] := Q(X[1])Wid(X[1])$;
$S := V[1]; K := 1$;
repeat
{
    find one of intervals $V[1], \dots, V[K]$ of the greatest width$(V[L])$;
    $X[L] := Lhalf(X[L]); X[K+1] := Rhalf(X[L])$;
    $V[L] := Q(X[L])\, Wid(X[L]); V[K+1] := Q(X[K+1])\, Wid(X[K+1])$;
    $K := K + 1$;
    $S := \Sigma\{V[K] | x = 1..K\}$
}
until Wid$(S) \leqslant w$ // Statement 2//;
```

Result. $s \in S$.

Remark 5.4. The value of S cannot be calculated by means of using preceding value of S:

$S := S_{prec.} - V_{prec.}[L] + V[L] + V[K+1]$ because of the following law for intervals: $Wid(A \pm B) = Wid(A) + Wid(B)$.

Hence, if the width of an interval B is positive then $Wid((A+B)-B) > \mathit{Wid}(A)$.

Remark 5.5. For given (very smooth) functions $q(x)$, the participant may venture to use any approximate formula for definite integrals (an estimation of a remainder term is practically impossible), but for given non-smooth functions it would be in vain.

Task 5.5. Given a polynomial $p(x,y)$ (such that $p(x,y) > 0$ for $|x| >> 1$ or $|y| >> 1$). Find an integer interval containing the area of the figure $F = \{(x,y) \in R^2 | p(x,y) < 0\}$.

Solution.

1st step. Build an interval representation $[P_-, P_+]$ for p.

2nd step. By means of a rough estimation find any interval vector $[a_-, a_+] \times [b_-, b_+]$ containing F (interval representation of F).

3rd step. The exhaustive summation (*Num* means the number of elements of the set):

$$\begin{aligned}
&\mathit{Area}(F) \in [Num\{(x,y)|P_+(x,x+1,y,y+1) < 0,\\
&\quad x = a_-..a_+ - 1, y = b_-..b_+ - 1\},\\
&(a_+ - a_-)(b_+ - b_-) - Num\{(x,y)|P_-(x,x+1,y,y+1) > 0,\\
&\quad x = a_-..a_+ - 1,\ y = b_-..b_+ - 1].
\end{aligned}$$

If the number $(a_+ - a_-)(b_+ - b_-)$ is too large then binary partition can be used.

Algorithm 5.4.

```
Define array Z[ ] of 2-dimensional interval vectors.
$Z[1] := [a_-, a_+] \times [b_-, b_+]$;
$A_- := 0;\ A_+ := (a_+ - a_-)(b_+ - b_-);\ K := 1$;
repeat
{find one of $Z[1], \ldots, Z[K]$ of the greatest width $(Z[L])$;
$P_Z := P(Z[L])$;
if $0 \in P_Z$ then
{$Z[L] := Lhalf(Z[L]); Z[K+1] := Rhalf(Z[L])$;
$K := K + 1$}
else
{{if $P_{Z+} < 0$ then $A_- := A_- + Area(Z[L])$
else
$A_+ := A_+ - Area(Z[L])$}};
$Z[L] := Z[K]; K := K - 1$}
}
until $Wid([A_-, A_+]) \leqslant w$ // Statement 2 //;
```

Result. $\mathit{Area}(F) \in [A_-, A_+]$.

6. Tasks of Discrete Content Arising from Continuity

The following types of tasks are of “common” discrete content but we did not meet such ones at informatics olympiads. They are solved by same methods.

Task 6.1. Given a polynomial $p(x)$ and two numbers $a < b$, find
$P_{\min} := \min\{p(x)|x = a..b\}$ (and find all $\{x \in a..b|p(x) = P_{\min}\}$).

Task 6.2. If N is even, find
$P_{\min} = \min\{p(x)|x\text{is integer}\}$ (and find all $\{\text{integer}x|p(x) = P_{min}\}$).

Task 6.3. Given a polynomial $p(x)$ and two numbers $a < b$, find

$$\{x = a..b - 1|p(x)p(x + 1) \leqslant 0\}.$$

Task 6.4. Given a polynomial $p(x)$, find all
$\{integer x|p(x)p(x + 1) \leqslant 0\}$.

Remark 6.1. This task is not equivalent to the task of separating all (real) roots of a polynomial. For example, if $p(x) = 9(x - \frac{1}{3})(x - \frac{2}{3}) = 9x^2 - 3x + 2$ then $p(x) > 0$ for all integer x.

Such tasks are more interesting for multi-dimensional cases where exhaustive search is evidently too slow. But a simple generalization may be incorrect. For example, the following task:

Given a polynomial $p(x, y)$, find
$\{\textit{integer}(x, y)|0 \in \textit{Outer}(\{p(x, y), p(x+1, y), p(x, y+1), p(x+1, y+1)\}$ is incorrect: it has infinitely many solutions for $p(x, y) = x + y$.

Task 6.5. Given a polynomial $p(x, y)$ (such that $p(x, y) > 0$ for $|x| >> 1$ or $|y| >> 1$) and natural M. Find *Num*{$p(x, y)$ *<0 | x and y are integer}* (*mod M*).

7. Conclusion

We hope that using tasks of proposed above types would attract attention of computer science students and computer scientists to the problem of validation of common approximate caculations and would expand the scope of tasks on olympiads in informatics of various levels. Also, distinguishing constructive (i.e., realizable on computer) methods among all mathematical ones would be useful in forthcoming professional activity of contestants of olympiads.

Acknowledgements. I thank Ulan Degenbaev for correction of the text.

References

Moore, R.E. (1966). *Interval Analysis*. Prentice-Hall, Englewood Cliffs, N.Y.
Pankov, P.S. (1978). *Validating Computations on Electronic Computers* (in Russian). Ilim Publishsing House, Frunze, Kyrgyzstan. Review: MR 82a:65004.
Pankov, P.S. (1978). A combined method for proving certain theorems of mathematical analysis via a computer. *Cybernetics* (translation of *Kibernetika*, Institute of Cybernetics, Kiev, Ukraine), 14(3), 441–448.

Pankov, P.S., Dolmatov, S.L. (1979). Substantiable evaluations by electronic computers and their application to one problem in combinatorial geometry. *Information Processing Letters*, 8(4), 202–203.

Pankov, P.S., Bayachorova, B.D., Yugai, S.A. (1982). Numerical theorem proving by electronic computers and its application in various branches of mathematics. *Cybernetics* (translation of *Kibernetika*,), 18(6), 840–848. Review: MR84i:68158.

Pankov, P.S. (1987). Machine presentation of information on continuous objects. *Scientific-Technical Information*. Series 2: information processes and systems, 2, 17–23 (in Russian).

Pankov, P.S., Janalieva, J.R. (1996). *Computer Mathematics*, Part I (in Russian). International University of Kyrgyzstan, Bishkek.

Pankov, P.S., Altynnikova, L.A., Janalieva, J.R. (2002). *Computer Mathematics*, Part II (in Russian), International University of Kyrgyzstan & Osh State University, Osh.

Shanin, N.A. (1968). Constructive real numbers and constructive function spaces. *Translation of Mathematical Monographs*, Vol. 21, American Mathematical Society, Providence, Rhode Island.

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