Tasks in Informatics of Continuous Content

Pavel S. Pankov *(pps50@rambler.ru),*

(Kyrgyzstan)

Brisbane - 2013

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.

All tasks of types proposed below can be written without special terminology, for example:

Task 1. Given an integer number $N \in 10..10^{1000}$. Find an integer K that the solution of the equation $x^3+x=N$ fulfills the assertion $K \le x < K+1$. *(Or, without real numbers at all)* …that

$K^3+K \le N$ and $(K+1)^3+(K+1) > N$.

(!) 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.

We will consider continuous objects: real numbers (basic objects), vectors in R^n, polygons, polynomials, continuous functions etc. To make such definitions strict, they will be presented (in the task conditions) by means of integer numbers.

Since not all real numbers can be represented by any algorithm (by constructive methods), the following notion was introduced (Turing, Shanin): Any real x is said to be *computable* if there exists an algorithm which reworks any natural number n in such integer m that

$$|x - m/n| < 1/n.$$

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 (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.

Denote $Wid(X)= x_+ - x_-$ (the *width* of X); the width of an interval vector is the maximum of widths of its components.

The narrower X, the better the presentation.

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

$mid(X) := int\ (X_- + X_+)/2$;

splitting: $Lhalf\ (X) := [X_-, mid(X)]$,

$Rhalf\ (X) := [mid(X), X_+]$.

The generalizations of the last two operations are also applicable to interval vectors.

Outer interval presentation of finite or other bounded sets in R^n: *Outer(W)* is the “least” box (interval for $n=1$, interval vector for $n>1$) containing the set W. The simplest example:

$$Outer(\{a,b\})=[min\{a,b\}, max\{a,b\}].$$

Recall the obvious formulas of interval analysis:

$$[x_-, x_+] + [y_-, y_+] = [x_- + y_-, x_+ + y_+];$$

$$[x_-, x_+] - [y_-, y_+] = [x_- - y_+, x_+ - y_-];$$

$$[x_-, x_+] \cdot [y_-, y+] =$$

$$Outer(\{x_- y_-, x_+ y_+, x_- y_+, x_+ y_-\}).$$

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.

(!) 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.

Denote $P^*(x_-, x_+) = Outer(\{p(x)|\ x_- \le x \le x_+\})$. It is the best possible estimation but cannot be calculated practically. If $P(x_-, x_+) \supseteq P^*(x_-, x_+)$ then it is said an *interval representation* (or *minorant* and *majorant*) for the function $p(x)$. If $P_1(x_-, x_+) \supseteq P_2(x_-, x_+)$ then P_2 is said *not worse* than P_1.

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.

Correct statement of problems and scoring

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

Statement 1. Find a semiopen integer interval of width 1 containing the solution.

But the general task to obtain such integer interval is incorrect.

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

Statement 3. … of width not greater than a given natural number (greater than 2).

There are two ways to score an answer given by the contestant:

- the jury knows the narrowest integer interval and an answer is to contain it;

- the scoring program checks an answer by a posteriori computations.

Types of tasks

General task 1. Given a function in a domain $G \subset R^n$ f: $G \to R$; find an integer interval containing its least (greatest) value.

General task 2. Given an equation in a domain of type $f(x)=0$ $(x \in G)$, find an integer interval (all integer intervals) containing its least (greatest) solution (all solutions correspondingly).

General task 3. Given a geometrical object, find an integer interval containing any measure (length, area, volume) of it.

Acceleration of Algorithms

Most of tasks listed above can be transformed to the following type:

Given a (slow) algorithm. Write a program yielding the same result in appropriate time.

For instance, Task 6b:

u:=0; equal:=false;

Repeat

u:=u+1;

for x:=1 to u { for y:=1 to u { for z:=1 to u { for t:=1 to u

{if $x^5+y^5+z^5+t^5=u^5$ *then*

{equal:=true; x1:=x; y1:=y; z1:=z; t1:=t } }}}}

until equal;

Output x1,y1, z1, t1, u.

Examples of tasks

Task 2. Given $N>3$, a polynomial $p(x) := \Sigma\{A[n]x^n \mid n=0..N\}$ and two numbers $a < b$, find an integer interval containing $p_{min} := min\ \{\ p(x) \mid a \leq x \leq b\}$.

Example: $min\{6\ x^2 - 30\ x + 40 \mid -1 \leq x \leq 8\ \} \in [2, 3]$

(!) Such example is necessary; otherwise many contestants will take integer values of x only:

$min\{6\ x^2 - 30\ x + 40 \mid x = -1..\ 8\ \} = 4$.

The simplest algorithm of exhaustive search: *calculate*

$P_{min} := [min\ \{P\_(k, k+1) \mid k=a..b-1\},\ min\ \{\ p\ (k) \mid k=a..b\}]$.

More effective algorithms use bisectional search; one of them demanding the least volume of memory: successful proving of inequalities (lower bounds for p_{min}).

Task 3. If N is even and greater than 2, find an integer interval containing $p_{min}:=min\ \{p(x) \mid x \in R\}$.

Solution. Find a priori boundaries for *arg min p(x)*. Then the task is reduced to Task 2.

Consider General Task 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 4. If $A[0]<0;\ A[n]\geq 0$ ($n=1..N$), 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, Task 4. Given $N>2$, (large) natural K, a polynomial $p(x)$ ($A[0]>0$, $A[n]\geq 0 \mid n=1..N$) and two positive numbers $a<b$, find an integer interval containing the area S between the x-axis, the graph of the function $q(x):=K/p(x)$ and lines $x=a$ and $x=b$.

(!). We evade the term “integral”, cf. remarks above.

Task 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 S of the figure $F:=\{(x,y)\in R^2 \mid p(x,y)< 0\}$.

Acceleration of algorithms

Most of tasks listed above can be transformed to the following type:

Given a (slow) algorithm. Write a program yielding the same result in appropriate time.

For instance, Task 1:

input N; K := 0;

repeat $K := K+1$; $\{F:=K^3+K-N\}$ *until* $F<0$;

Output K.

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.

Thank you for attention!