Number Theory Problems – Differences by Grade

Kalvis Apsitis, Maruta Avotina

June 30, 2019

Abstract: *In this article we study Olympiad problems in number theory (NT) offered in Latvia and Estonia. We compare several characteristics of the problems depending on the grade - the age for which they were offered. We compare them by the skills involved, by their question types and concepts used in the source text of the problem.*

1 Introduction

Secondary schools and gymnasiums in Latvia and Estonia share many similarities.

- Lower level of the secondary-school is Grades 7-9 (ages 14-16), upper level is Grades 10-12 (ages 17-19).
- Most NT concepts (divisibility, remainders, prime numbers, LCM and GCD, decimal notation) are already covered in Grades 5-6, so the theory part in NT mostly stays the same for all Grades 7-12.
- Olympiads in Latvia and in Estonia start with the regional tour followed by the national tour (only the highest scoring students from the regional tour enter the national one). There is also the open tour (in Estonia 2 open tours per year), where everyone can participate.
- Olympiad papers in Latvian and Estonian Olympiads have 5 problems for 4-5 hours. On average, each Olympiad paper has one NT problem (others are on geometry, algebra, combinatorics and also “school problems”, where the standard math curriculum should be sufficient.)
- Problems for Grades 7, 8, 9, 10, 11, and 12 do not intersect. In Estonian Open Olympiads there are only two age groups: up to Grade 10 or Grades 11-12; in this study both were considered “senior” (Kosik, 2019))

Problem selection committee in Latvia has some guidelines, how to assign NT problems:

1. Junior-level problems should not refer to geometrical progressions, exponential functions or logarithms, or floor function $\lfloor x \rfloor$.
2. Junior-level problems do not use ellipsis notation (omitted subexpressions with dots: $F(1) + F(2) + \cdots + F(n)$ or $\sum_{k=1}^{n} F(k)$).
3. A problem using the “regular” rule of divisibility by 9 is fine for all Grades 7-12, but a generalization (each number n is congruent to the sum of its digits modulo 9) is only used for Grades 10-12.

The goal of this paper is to find empirical evidence of differences caused by conscious or unconscious choices of the problem selection committees in both countries. This would help to plan Olympiad training curricula for different age groups.

2 Description of the study

2.1 The Data Set

We analyzed regional, national and open Olympiads in Estonia and Latvia split into two age groups: Junior (Grades 7-9) and Senior (Grades 10-12). Competitions during the years 2010-2019 were used. The total number of number theory problems per country and age group:

- Estonian junior-level NT problems: 59
- Estonian senior-level NT problems: 115
- Latvian junior-level NT problems: 159
- Latvian senior-level NT problems: 107

Latvian Open Olympiads (NMS, 2018) and Regional Olympiads start at Grade 5, but we only annotated NT problems for grades 7-12. Certainly, there are many activities in Latvia and Estonia that were not analyzed, such as correspondence competitions or team Olympiads. The 440 problems analyzed for this article should be representative enough to draw some conclusions.

2.2 Annotations of Strategies vs. Skills

In order to solve an Olympiad problem (namely, a task not having a predefined solution procedure) two steps are typically needed:

1. Select an overall strategy, design a plan to solve the problem. For more complicated problems several intermediate results may need to be formulated with several substrategies.
2. Apply steps (and NT subject-matter knowledge) to implement the selected strategy.

Paul Zeitz (2007, page 22) mentions several strategies *penultimate step*, *get your hands dirty*, *wishful thinking*, and *make it easier*. In NT problems there are a few more - *experimentation*, *mathematical induction*, *proof from the contrary* along with some other patterns that affect the high-level structure of the solution: analyzing the extreme element, using symmetry, etc. Strategies appear in our annotated database, but are not used in this article.

Skills are needed to execute the strategy selected previously. Multiple ways to solve a problem would likely lead to different lists of skills. Even for the same problem assigning skills is somewhat subjective; it may depend on the understanding or paradigms of the person who created the annotation.

2.3 Annotation of Question types vs. Genres

Most Olympiad problems in geometry can be formulated as follows: "Prove that for all shapes *A*, *B*, … satisfying condition *X*, the condition *Y* also holds." In our NT problem annotations this would be marked as **Prove.ForAll*, but there are other question types as well. Question type is evident from the problem source text, it determines which strategies can solve it, and how the solution should look.

- **Find.All** – Find all objects satisfying the condition and prove that there are no others. The objects may be anything.
- **Find.Count** – Find the total number of objects satisfying the condition.

- **Find.Optimal** – Find the largest or the smallest number satisfying the condition. The solution should have two parts - first find some answer, then prove that it is the optimal one.
- **Find.Only:** – A numeric example, comparison of two large numbers or other computational task, where it is obvious that the answer exists and it is unique.
- **Find.Any** – find any object satisfying the condition.
- **Prove.ForAll** (**Prove.NotExists**) - Prove that a general statement holds (or that its negation cannot exist).
- **Prove.Exists** – Prove that an object satisfying a condition exists. Requires an example (olympiad problems may have non-constructive proofs of existence - see **Example** below).
- **Prove.Other** – In these problems there are alternating layers of quantifiers "for all" and "exists".
- **ProveDisprove.Exists** – The hypothesis states that something exists; the solution should either prove it (by providing an example) or disprove (by a general statement).
- **ProveDisprove.ForAll** – The hypothesis is formulated as a general statement. One should prove that general version or disprove by providing a counterexample.
- **ProveDisprove.Other -** The hypothesis has more than two alternate quantifiers ("for any *x* there exists *y*" or "there is an *x* such that for all *y*", or "the strategy should work regardless of the moves of your adversary", or "there are infinitely many *x* satisfying condition" and so on).
- **Algorithm** The solution should provide a procedure or a method of construction and prove that it is valid. In the **Example** below, the winning strategy for the 1st player should be provided.

Example: Two players make alternate moves: In each move a player writes a positive integer between 1 and 8 (inclusive). It is forbidden to write a number which divides any number written previously. The player who cannot make a move loses. Show, how the 1st player can always win.
(*Latvian Open Olympiad, 2002. Grade 7, Problem 4.*).

Note: If we change the above problem just a little by asking to prove that the 1st player always wins (without asking to show **how** she/he can win), then the question type would switch from **Algorithm** to **Prove.ForAll** and a much shorter solution would be possible:

(1) It is a finite combinatorial game, so exactly one player should have a winning strategy;
(2) Assume that the 1st player has a winning strategy and it starts with number 1. In this case we are done;
(3) Or else the 2nd player can win by responding with $N > 1$ to the 1st player's move 1. In this case the 1st player can start by N and emulate the winning strategy of the 2nd player.
Our database in (NMS, 2019) includes many more examples for each question type mentioned above.

In number theory there are also some narrower categories of Olympiad problems we could call genres. They share the question type, but they also have similar statement of the problem condition. For example:

- Solve an equation in integer numbers (or positive integers)
- Find which player has a winning strategy in an NT game played on a blackboard
- Prove that two statements are logically equivalent
- Write an explicit formula for an integer sequence

- Fill in numbers in a table (or circles in a diagram) to satisfy some condition

Our annotations point to the genres whenever they are known, but this classification is not yet comprehensive, so it is not used in the age-based comparisons. About a third of Olympiad problems annotated in our database do not fall in any typical genre; many small or artificial “genres” would be needed to classify all of them.

2.4 Annotation Scheme

Each problem can receive the following mandatory annotations:

- **country** (“EE” or “LV”), **Olympiad identificator** (the country-specific acronym indicating, if it is regional, national, open competition), **year**, **grade** - We create IDs to locate the NT Olympiad problems. Example 1 is identified as **LV.AO.2002.7.4** (Latvia, “Atklātā olimpiāde (AO)” = Open, Year 2002, Grade 7, Problem 4).
- **strategy** - one or more general methods to solve the problem (see (Polya, 1945) and (Zeitz, 2007) for examples of problem solving strategies). We would like to come up with an authoritative list of strategies, but do not have a proper reference yet.
- **skills** - ability to perform certain steps or tasks that are part of the general strategy (unlike strategies, skills may be NT specific). Skills are ordered in a tree-like hierarchy; upper levels of the tree are shown in Section 3.1.
- **question type** - what is the required result. Values are listed in Section 2.3.
- **concepts** appearing in the problem (for example, problems mentioning prime numbers, greatest common divisors, floor function). Any well-defined mathematical concept may appear, if it is used in the statement of the problem. If the solution of the problem uses graphs, but graphs are not necessary to understand the problem, then these are marked as **skills**, not **concepts**.

Annotated database also has some optional attributes:

- **genre** a narrower problem category that often implies the question type (such as integer equations, analyzing adversarial games, filling a table). This can be used to filter out all integer equations, all games, etc.
- **seeAlso** pointers to related olympiad problems and other math education resources.

3 Comparison between the Junior and Senior Levels

3.1 Skills

To comply with the best practices in the Instructional Design, every skill or task is described in a single phrase starting with a verb (Stolovitch & Keeps, 2011). The list is not meant to be exhaustive (in fact, there are many more skills used in Estonian and Latvian NT problems, many of them applicable outside the NT). Moreover, each skill in this list can be expanded into several more basic/specific skills. Our annotated NT problem database (NMS, 2019) specifies skills at a more granular level.

In this aggregate view we want to show a high-level distribution. Each skill is followed by two numbers **A/B**. The first number shows, how many junior-level problems use skills from this group, the second number shows the number of senior-level problems (from our data set of 440 problems).

Algebraic skills

- 1.1. Use algebraic identities: **13/11**
- 1.2. Reorder and compute sums: **7/3**
- 1.3. Use the properties of polynomials with integer coefficients: **2/3**
- 1.4. Manipulate rational fractions: **0/2**
- 1.5. Manipulate linear equations and systems: **5/2**
- 1.6. Prove and use inequalities: **16/9**

Divisibility skills

- 2.1. Use divisibility relation and its axioms: **15/2**
- 2.2. Use prime factorization: **13/18**
- 2.3. Use the properties of LCM and GCD: **7/4**
- 2.4. Compute and use p-valuations: **0/2**

Modular arithmetic skills

- 3.1. Use parity and the remainder expression $a = dq + r$: **7/10**
- 3.2. Add, subtract and multiply congruencies by any module: **0/14**
- 3.3. Use inverse elements by prime modules, the periodicity of $a^x (mod\ p)$: **0/3**
- 3.4. Solve some congruence equations: **0/8**

Decimal notation skills

- 4.1. Make conclusions and estimates about decimal representations: **12/5**
- 4.2. Use divisibility rules: **13/19**
- 4.3. Express digit manipulations with algebra: **4/4**

Integer sequence skills

- 5.1. Define sequences and use properties (monotone, bounded, periodic): **0/3**
- 5.2. Use arithmetic progressions: **2/13**
- 5.3. Use geometric and other progressions defined explicitly or recurrently: **1/4**
- 5.4. Estimate sequence growth and the gaps between their members: **4/3**
- 5.5. Find sequences satisfying certain properties: **0/6**

3.2 Question Types for Junior/Senior Levels

As before, the first number is the count of junior-level problems, the second number is the count of senior-level problems belonging to the question type. Some problems have multiple items (asking different kinds of questions), so the categories do not add up to the number of problems themselves.

- Computations (**Find.Only**): **6/9**
- Finding an example (**Find.Any**): **11/1**
- Finding best example (**Find.Min**, **Find.Max**): **20/18**
- Counting possibilities (**Find.Count**): **4/10**
- Find all solutions (**Find.All**): **62/40**
- Proof by example (**Prove.Exists**): **1/3**

- Proof for the general case (**Prove.ForAll**): **9/27**
- Proof of impossibility (**Prove.NotExists**): **2/5**
- Prove or disprove an existence (**ProveDisprove.Exists**, **ProveDisprove.ForAll**): **23/28**
- Prove or disprove more advanced (**Prove.Other**, **ProveDisprove.Other**): **3/11**
- Algorithms, game strategies (**Algorithm**): **5/3**

There is also a major difference between Estonian and Latvian problem sets regarding the question type **Find.All**. In Estonian Olympiads there are 81 problems, but in Latvian Olympiads just 21 (the overall data sets are similar size - see Section 2.1) This is probably explained by the predominance of integer equations in Estonian Olympiads.

3.3 Concepts

The "Reports" section in annotated database by Correspondence School of Mathematics (NMS, 2019) contains a bar-chart with some popular concepts in the NT problems. Near the top there are concepts found predominantly in junior problems (Grades 7-9), near the bottom we display the concepts that are predominantly or exclusively in senior problems (Grades 10-12).

4 Conclusions

1. NT problems for senior levels (Grades 10-12) use a somewhat broader set of skills, but the difference
2. Question types and genres differ more strongly by country (Latvia vs. Estonia) than by age. They may reflect certain "national traditions". Namely, what formulations are preferred by the problem selection committees in both countries.
3. Even junior level NT problems ask questions with alternating quantifiers ("for any *x* there is an *y*", proofs that certain sets are infinite).
4. The most reliable difference between junior-level and senior-level problems was observed regarding the concepts used in the problem.

Some topics not covered in this paper, but could build on the annotated database of NT Olympiad problems:

- Metadata for the Baltic Olympiad problems should be made available via public API - application programming interface (in English) to enable uniform access to this information for other types of research. As with any statistics project our results should become fully transparent and repeatable.
- Correlations between the subject-matter skills and the observed results from the national competitions. For example, one could compare, if NT problems based on the divisibility rule by 9 are easier or harder compared to NT problems using the divisibility rule by 11.
- To make task/skill annotations more objective, it would be beneficial to compare how several people (math teachers, Olympiad problem graders, etc.) analyze the same problem into skills. For this we would need an interactive annotation tool.

References

Kosik, O. (2019) Estonian Mathematical Olympiads. Retrieved in June 2019. URL: http://www.math.olympiaadid.ut.ee/html/

NMS (2018) Archive of Problems by Correspondence School of Mathematics (Neklātienes Matemātikas Skola, NMS, in latvian). Retrieved in June 2019. URL: http://nms.lu.lv/uzdevumu-arhivs/latvijas-olimpiades/

NMS (2019) Olympiad Problems in Number Theory: Annotated Database by Correspondence School of Mathematics (Neklātienes Matemātikas Skola, NMS, in latvian). Retrieved in June 2019. URL: http://www.dudajevagatve.lv/numtheory/prob/

Polya, G.(1945) *How to Solve It.* Princeton University Press

Stolovitch, H., D., Keeps, E., J. (2011) *Telling Ain't Training*, 2nd Edition. Association for Talent Development

Zeitz, P. (2007) *The Art and Craft of Problem Solving*, 2nd Edition. John Wiley and Sons