International Journal of Mathematical Education in Science and Technology

ISSN: 0020-739X (Print) 1464-5211 (Online) Journal homepage: http://www.tandfonline.com/loi/tmes20

Likert-scale questionnaires as an educational tool in teaching discrete mathematics

O. A. Ivanov, V. V. Ivanova & A. A. Saltan

To cite this article: O. A. Ivanov, V. V. Ivanova & A. A. Saltan (2018): Likert-scale questionnaires as an educational tool in teaching discrete mathematics, International Journal of Mathematical Education in Science and Technology, DOI: 10.1080/0020739X.2017.1423121

To link to this article: https://doi.org/10.1080/0020739X.2017.1423121

Published online: 12 Jan 2018.

INTERNATIONAL JOURNAL OF MATHEMATICAL EDUCATION IN SCIENCE AND TECHNOLOGY, 2018
https://doi.org/10.1080/0020739X.2017.1423121

Taylor & Francis
Taylor & Francis Group

CLASSROOM NOTE

Check for updates

Likert-scale questionnaires as an educational tool in teaching discrete mathematics

O. A. Ivanov[a], V. V. Ivanova[b] and A. A. Saltan[b]

[a]Department of General Mathematics and Informatics, St. Petersburg State University, St. Petersburg, Russia; [b]Department of Information Systems in Economics, St. Petersburg State University, St. Petersburg, Russia

ABSTRACT
In this paper, we report on the results of an experiment in teaching discrete mathematics to students majoring in business informatics. We supplemented our problem-based approach to teaching the course with a set of Likert-scale surveys or questionnaires that helped improve the students' performance. On the one hand, these surveys gave us feedback and, on the other, encouraged the students to reflect on the subject-matter. The experiment was quite successful, as the grades obtained by the students on the exam were significantly higher than usual. Here, we describe the structure of the surveys and the method of evaluation of the experimental results.

ARTICLE HISTORY
Received 5 October 2017

KEYWORDS
Active learning; Likert-scale questionnaires; discrete mathematics; student performance

1. Introduction

A course in discrete mathematics is in fact of an interdisciplinary nature. It develops the analytical, logical, and algorithmic thinking necessary for successful study of many other disciplines [1,2]. Although such a course is essential for students majoring in business informatics, like other interdisciplinary courses, it appears to be quite challenging for them [3].

Approaches to designing the structure of interdisciplinary courses and teaching them in such a way as to both make them more attractive to students and improve their performance have received much attention in research and practice over the last decade. Various approaches assuming the involvement of students in the learning process with the purpose to obtain long-lasting and meaningful learning are nowadays known as active learning. Active learning suggests students' engagement in such high-order activities as analysis, synthesis, and evaluation [4]. There is good evidence that the implementation of an active learning approach improves students' performance without noticeable loss in academic knowledge (see, e.g. [5–7]). The types of active learning interventions considered vary widely. As shown in [8], in teaching physics, the implementation of the technique called 'interactive lecture demonstrations' yields significant improvements in learning outcomes. On the other hand, there is evidence that the incorporation of a programming practicum in a discrete mathematics course also promotes more active learning [9].

CONTACT O. A. Ivanov o.a.ivanov@spbu.ru

In our discrete mathematics course as formerly taught, we had incorporated elements of the active learning approach from the very beginning, and the lectures were supplemented with problem-solving sessions and a programming practicum. However, the students' results (in particular, their grades) were far from what one might desire.

The starting point of our experiment came from an observation made in [2]. The students were asked to specify on a so-called Likert scale (ranging from *strongly agree* to *strongly disagree*) their level of agreement or disagreement with the following statement: 'The reason why it is hard to decipher a message encoded by the RSA method lies in the fact that the complexity of finding the inverse to a given element of $\mathbb{Z}_k$ depends exponentially on the number k'.

Since the suggested reason is in fact wrong, the correct reaction to this statement would be to check 'strongly disagree' on the Likert scale. Nevertheless, more than half of the students agreed with the statement. Such a reaction shows clearly that the students had grasped neither the central idea of the RSA encryption algorithm (the difficulty of factoring 'very large' numbers that are products of two primes), nor the algorithm for finding the inverse for a given element in the ring $\mathbb{Z}_k$. In fact, the complexity of finding $a^{-1} \in \mathbb{Z}_k$ depends linearly even on a.

Thus, the reaction of the students to the given statement provides an indication of the level of understanding (or rather misunderstanding) of the topic under study. The novelty of our proposed approach was to supplement the standard approach to learning, that is, one including home and in-class assignments as well as a mid-term colloquium, with Likert-based surveys. These surveys were followed by a discussion of the main topics and were intended to 'close the knowledge loop' in the sense of providing both students and lecturer with information regarding the level of understanding of the topic. In this paper, we describe the results of such an experiment conducted with freshmen majoring in business informatics. Our main goal was to improve our students' conceptual understanding of discrete mathematics.

2. Description of the experiment

2.1. Design and scheduling of the experiment

The experiment aimed to investigate the extent to which the implementation of Likert-scale surveys followed by detailed discussion can contribute to a better understanding of discrete mathematics.

The experiment was carried out in the Spring of 2017 with 22 students majoring in Business Informatics as participants. Four surveys were conducted: the first three were oriented towards theory, and the fourth to the implementation of algorithms using CAS Wolfram Mathematica. In each of the four questionnaires provided, we asked the students to express their attitude toward the statements on the following Likert scale

strongly agree	*agree*	*hard to say*	*disagree*	*strongly disagree*

The first three questionnaires each consisted of 20 statements and the fourth questionnaire of 14. Each questionnaire contained an equal number of true and false statements (or 'wrong-worded statements', so-called).

Here is a detailed list of topics mentioned in the questionnaires:

Survey 1 (March): sequences and recurrence relations; sets and counting; combinatorics.

Survey 2 (April): modular arithmetic and some applications; the main algorithms relating to binary heaps and binary search trees; sorting algorithms.

Survey 3 (the beginning of May): the main algorithms relating to graphs and trees; greedy algorithms relating to weighted graphs; relations on sets.

Survey 4 (the end of May): the syntax and semantics of operators and the implementation of the main algorithms of discrete mathematics in CAS *Mathematica*.

A final survey (similar to those described in [2]) was conducted just before the examination. Discussion of the results of each of the surveys took place a few days later.

Surveys 1–3 lasted 15 minutes each, and their discussions 15–20 minutes each. Survey 4 required a little more time (about 30 minutes), and its discussion took also 15–20 minutes.

As will be shown below, the surveys covered a wider range of concepts, ideas, and algorithms of the discrete mathematics course than did the regular quizzes. Also, since the restricted time forced the students to answer the questions without hesitation, their responses reflected their understanding of the course. A student's results on the surveys did not affect his/her final grade for the course, so he/she was motivated to provide answers deemed by them to be correct in the light of their knowledge and understanding.

2.2. Statements examples

Here are the first 10 statements of the first questionnaire and the first 5 statements of the last questionnaire.

Statements of questionnaire 1 (Here statements 2, 3, 6, 8, and 9 are the true ones.)

(1) The magnitude of a Fibonacci number is a quadratic function of its index.
(2) The set of all arithmetic progressions is the same as the set of all sequences satisfying a second-order linear recurrence relation.
(3) The number of combinations is a special case of the number of permutations with repetition allowed.
(4) The sum of the squares of the first n natural numbers is a quadratic polynomial in n.
(5) If the height of a tower of Hanoi is doubled, then the number of moves required to transfer the entire stack to another peg is approximately four times greater.
(6) If finite sets A and B have the same number of elements, then the number of injections $A \to B$ coincides with the number of surjections $A \to B$.
(7) In every set, the number of all 4-element subsets is greater then the number of all 3-element subsets.
(8) If a set has more than 3 elements, then the number of permutations on its elements is greater than the number of its subsets.
(9) The number of subsets of a finite set A coincides with the number of all maps from A to a 2-element set.
(10) If 13 out of 25 schoolmates like to read books and 17 adore computer games, then exactly 5 of them like both to read books and play on the computer.

Statements of questionnaire 4 (Here statements 1, 2, and 3 are true.)

(1) The procedure described below removes duplicates from a given list while preserving the order of the last occurrences of the items in the list.

```
nodupl[x_] := Module[{y = x, i = Length[x]},
  While[i > 0, {y, i} = {Delete[y, Most[Position[y, y[[i]]]]],
     i - Length[Most[Position[y, y[[i]]]]] - 1}]; Return[y]]
```

(2) Though the complexities of the bubble-sort algorithm and the insertion sort algorithm are both of the order $O(n^2)$, 'bubble-sort' is, on average, more time-consuming than 'insertion sort'.

(3) The procedure given below constructs the adjacency matrix of a graph from the adjacency list of the graph:

```
fromadj[m_] := Module[{a, n = Length[m]},
  a = Table[0, {i, 1, n}, {j, 1, n}];
  Do[a[[i, m[[i]]]] = 1, {i, 1, n}]; Return[a]]
```

(4) Execution of the following code yields a list of the first ten Fibonacci numbers:

```
fib1[n_] := fib1[n - 1] + fib1[n - 2];
fib1[1 _] = 1; fib1[2 _] = 1;
Table[fib1[n], {n, 1, 10}]
```

(5) The execution of the operator `digits[13]` results in the list `{1,1,0,1}` consisting of the binary digits of the number 13.

```
digits[n_] := Module[{s = {}, k = n, t},
  While[k ≠ 0, t = Mod[k, 2]; s = Append[s, t]; k = (k - t) /2];
  Return[s]]
```

2.3. *Discussion of the above statements*

We first consider the statements of questionnaire 1 in order.

Those agreeing with statement 1 either do not know *Binet's formula* or fail to understand its connection with the statement. That formula implies that $F_n \approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$, so that F_n depends *exponentially* on n.

Of course, statement 2 is true since, by the definition of an arithmetic progression, we have $x_{n+1} - x_n = x_n - x_{n-1}$, whence $x_{n+1} = 2x_n - x_{n-1}$. It is clear that a sequence satisfies the latter recurrence relation if and only if the sequence is an arithmetic progression.

Assume that an n-element set contains k elements of type 1 and $n-k$ elements of type 2. Then, the formula for the number of permutations with repetitions takes the form $\frac{n!}{k!(n-k)!}$, which is the number of k-combinations from a given n-element set. Note that, of course, it is not actually necessary to use specific formulas for these numbers. One can use instead the following purely combinatorial argument: There are $\binom{n}{k}$ ways to choose k places for objects of type 1 in a permutation of length n, with repetition. Students who *remembered* the proof of the formula for the number of permutations with repetition should easily be able to respond correctly to Statement 3.

We proved in class that the sum of squares of the first n natural numbers is a cubic polynomial in n. However, the fact that statement 4 is false is also clear from general considerations.

Statement 5 is completely wrong. Indeed, by the well-known formula, we need $2^4 - 1 = 15$ moves for a tower of 4 discs, while we need $2^8 - 1 = 255$ moves for a tower with 8 discs. Students agreeing with Statement 5 have failed to understand that the number of moves in the Hanoi problem depends exponentially on the number of discs.

Statement 6 is true because if two finite sets A and B have the same number of elements, then a map f: $A \to B$ is an injection if and only if f is surjection.

It would seem that the number of 4-element subsets of a set is greater than the number of its 3-element subsets. However, this is true only if the set contains more than 8 elements. Otherwise statement 7 is not true, since a 4-element set has only one 4-element subset, but four 3-element subsets. Thus, Statement 7 is false.

Statement 8 is the only statement where any calculation is required. If a set has 4 elements, then the number of permutations of its elements is $4! = 24$, while the number of its subsets is $2^4 = 16$. One then argues by induction as follows. If $n! > 2^n$, then $(n+1)! = (n+1) \cdot n! > 2 \cdot n! > 2 \cdot 2^n = 2^{n+1}$. Hence, Statement 8 is true since, by assumption, the number of elements in the given set is greater than 3.

Statement 9 is true because there is a one-to-one correspondence between the set of all subsets of a given set A and the set of all maps from A to the 2-element set $\{0, 1\}$.

Statement 10 would be true under the assumption that every student likes to engage in at least one of the two activities. Note that if we reword the statement by writing 'at least five' instead of 'exactly five', we get a true statement. Of course, Statement 10 is related to the inclusion–exclusion formula.

Note that the above 10 statements cover most of the topics included in Survey 1.

We now turn to the statements of questionnaire 4.

Statement 1 is true. The code does remove duplicates from a given list while preserving the order of the last occurrences of the items in the list. For example, applying the procedure `nodupl` to the list $\{1, 2, 3, 6, 6, 3, 5, 2, 1\}$, we obtain the list $\{6, 3, 5, 2, 1\}$. A similar procedure was considered in class, the only difference being that that procedure preserved the order of the *first* occurrences of the items in a list.

Statement 2 is also true because, on average, the bubble-sort algorithm requires twice as many comparisons as the insertion sort. Moreover, the number of renumberings in the bubble-sort depends quadratically on the length of the list, while for the insertion sort the dependence is linear. The difference in the execution speed was demonstrated to the students in the computer lab.

Of course, Statement 3 is true. We deliberately changed the name of the procedure to compel the students to think a little.

Table 1. Students' response to Questionnaire 1.

Statement	1	2	3	4	5	6	7	8	9	10
Right reaction	16	4	12	12	13	9	16	12	11	15
Wrong reaction	5	8	7	7	8	11	3	8	5	5
Neutral reaction	0	9	2	2	0	1	2	1	5	1

Statement 4 is false. The given code generates a list consisting of 10 ones. This question tests the students' knowledge of the CAS *Mathematica* syntax. The fact is that the form `fib1[1_]=1` of the assignment operator entails that *each* value returned by `fib1` is 1.

Statement 5 is also false. Though the given code indeed generates a list of binary digits of the number 13, the digits will be written in a different order: the execution of the operator `digits[13]` results in the list {1, 0, 1, 1}. The students should understand that the digits are calculated starting from the least significant digit. Since the operator `Append` appends the next element to the end of the list, the binary digits of 13 will be written in reverse order.

Thus, we see that these five statements test not only for knowledge of the syntax, but also for knowledge of procedures, theoretical results, and ability to read codes.

3. Experiment evaluation

3.1. Students' response

The Likert scale has been widely used in various psychometric studies, including research in mathematics teaching. For example, there are many papers devoted to determining the relation between mathematics instructors' beliefs and practices (see e.g. [10–12] and references therein). In such studies, it was, of course, necessary to take into account the difference between the answers 'agree' and 'strongly agree'. In our analysis of responses, however, we did not distinguish between agreement and strong agreement nor between disagreement and strong disagreement. We were concerned exclusively with basic understanding of the subject: students should agree with a correct statement and disagree with a wrong one. But we wanted to take into account the fact that students coming fresh to university often lack self-confidence so tend not to give categorical answers. Another goal of the surveys was to increase the students' interest in the subject under study, and, in this respect, the Likert scale was useful since it increases the emotional component of the learning process.

In Table 1, we give the numerical breakdown of the responses of the 21 students to the statements of Questionnaire 1, including the numbers of 'undecides'.

This table clearly provides much material for discussion. The poor overall response to Statement 2 was easily explained by the fact that the idea of defining an arithmetic progression by means of a recurrence relation had not been introduced in class: it was thus left to the students to understand on their own that an arithmetic progression could be given via a second-order recurrence relation. On the other hand, the fact that less than a half of the students correctly understood Statement 6 indicates that they had not understood in depth the concepts of injection and surjection. The fact that only a slim majority of students understood Statements 3 and 4 would seem to indicate that, at that time, they were having trouble understanding the definition of permutations with repetitions and did not

Table 2. Students' response to Questionnaire 4.

Statement	1	2	3	4	5
Right reaction	9	10	12	6	14
Wrong reaction	9	8	6	13	5
Neutral reaction	1	1	1	0	0

Table 3. Students' response to the last Questionnaire.

Statement	1	2	3
Right reaction	17	16	17
Wrong reaction	2	3	3
Neutral reaction	2	2	1

remember the formula for the sum of squares of the first n positive integers (and, moreover, did not remember the basic idea for deriving formulas for sums of exponents of these numbers).

The results of Questionnaire 4 are as shown in Table 2.

The fact that two-thirds of the students responded incorrectly to Statement 4 indicates that they do not understand the syntactic structure of the definition of new operators in CAS Mathematica. The responses to Statement 1 demonstrate that large number of students are not able to understand a program code.

We discuss the pedagogical conclusions of our experiment in the next section; for now we just give one example involving the bubble-sort algorithm.

It can be seen from Table 2 that when answering Questionnaire 4, only half of the students understood that this algorithm is less efficient than the insertion sort algorithm. It is interesting to consider the change between the time this Questionnaire was administered and the final exam. To this end, the following statements were included in the last (pre-examination) survey:

Statement 1. The bubble-sort algorithm is one of the most effective methods for sorting large lists.

Statement 2. Applying this algorithm to a list consisting of 10 elements, in a worst case scenario, one has to perform slightly fewer than 100 comparisons and the same number of transpositions of pairs of elements in the list.

Statement 3. Doubling the number of elements of a list increases the time for sorting this list by the bubble-sort algorithm approximately by the factor 4.

Table 3 indicates that by the end of the experiment, the majority of students had understood that the complexity of the bubble-sort algorithm depends quadratically on the length of the list, that the greatest number of comparisons needed for sorting a list consisting of n elements by this algorithm is approximately equal, not to n^2, but to $n^2/2$, and that this algorithm is in no way the most effective.

3.2. *Student's performance*

In evaluating the results of our project, we used both objective and subjective data. The objective data consisted in the distribution of students' scores on the discrete mathematics exam, as compared with the distribution of scores of students who sat the exam in 2016. (We

Table 4. Distribution of students' examination scores.

	0–16	17–32	33–48	49–64	65–80	81–100	Mean
Spring 2016	10%	10%	30%	45%	0%	5%	48.8
Spring 2017	5%	10%	15%	45%	15%	10%	57.4

Table 5. Average examination scores in Calculus and Discrete Mathematics courses.

	Spring 2016	Spring 2017
Calculus	44.2	46.4
Discrete math	48.8	57.4

Table 6. Students' feedback on the experiment.

Didn't affect at all	Didn't affect	Don't know	Affected	Strongly affected
1	2	0	13	5

were unable to use exam results from previous years because of differences in the course material.) Table 4 shows the percentage distribution of students' scores across the indicated intervals as well as the average score.

One sees an obvious improvement in the results: the average examination scores improved by 8.6 percentage points, a relative increase of more than 17%.

We did not have the opportunity to form a *control* group of students. Instead, as a next best substitute, we compared our results with those obtained in the final examination in Calculus, taught in the traditional way. Table 5 shows the average scores obtained by students in that exam compared with their average score in discrete math.

(The average score of students enrolled a year earlier in the examination in discrete mathematics in Spring 2015 was 44.0.)

If we compare the ratio of the average scores obtained by students in the examinations in discrete mathematics and Calculus in Spring 2016 and Spring 2017, we see that the difference between the average scores for 2017 and 2016 is definitely related to the methodology used in teaching discrete mathematics in 2017. It follows from the above data that the ratio increased by 12%, which is twice the value specified in [7].

The subjective data in question were the results of an anonymous survey conducted after the examination. Students were asked to rate the extent to which their participation in the experiment *positively* affected their understanding of the discrete mathematics course. Table 6 presents the distribution of students' answers. (Note that one student had dropped out of the project).

Thus, the majority of students (86%) reported a positive impact of the methodology used. Since this methodology has proven its effectiveness, we will introduce it into the educational process in the future, conducting similar surveys online.

4. Concluding remarks

We have here presented our first experiment involving the use of a Likert-scale survey in a discrete mathematics course. The effectiveness of this approach lies in the fact that it promotes active learning among students. Discussion of their responses, both in small groups and in the classroom, leads to a better understanding of the course material on their part.

At the same time, the fact that a student's answer to a question formulated in an intriguing way suddenly turns out to be incorrect may well increase his/her interest in the course and desire to learn. Informally, our model of active learning might be depicted as follows:

$$\text{Understanding} \Rightarrow \text{Response} \Rightarrow \text{Reflection \& Discussion} \Rightarrow \text{Understanding}$$

In conclusion, we wish to note that the proposed pedagogical approach was also implemented, though in a more limited fashion, in the teaching of a databases course for second-year students. Within the course, only one survey (with subsequent discussion) was conducted. However, more than 60% of students noted that even that single questionnaire contributed significantly towards their understanding of the course.

Acknowledgments

The authors are deeply grateful to Robert G. Burns who kindly agreed to edit this manuscript.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] Page RL. Software is discrete mathematics. 8th ACM SIGPLAN International Conference on Functional Programming. Uppsala (Sweden): ACM Press; 2003. p. 79–86.
[2] Ivanov OA, Ivanova VV, Saltan AA. Discrete mathematics course supported by CAS Mathematica. Int J Math Educ Sci Technol. 2017;48:953–963.
[3] Hämäläinen W. Problem-based learning of theoretical computer science. Proceedings of Frontier in Education Conference. Savannah (GA): IEEE Press; 2004.
[4] Bonwel C, Eison J. Active learning: creating excitement in the classroom. Washington (DC): School of Education and Human Development, The George Washington University; 1991. (ASHE-ERIC higher education reports).
[5] Kim K, Sharma P, Land SM, et al. Effects of active learning on enhancing student critical thinking in an undergraduate general science course. Innov Higher Educ. 2013;38(3):223–235.
[6] Prince M. Does active learning work? A review of the research. J Eng Educ. 2004;93(3):223–232.
[7] Freeman S, Eddy SL, McDonough M, et al. Active learning increases students performance in science, engineering, and mathematics. Proc Natl Acad Sci. 2014;111:8410–8415.
[8] Georgiou H, Sharma MD. Does using active learning in thermodynamics lectures improve students' conceptual understanding and learning experience? Eur Phys J. 2015;36:015–020.
[9] McMaster K, Anderson N, Rague B. Discrete math with programming: better together. 38th SIGCSE Technical Symposium on Computer Science Education; 2007; Covington, Kentucky, USA: ACM Press. p. 100–104.
[10] Langrall CW, Thornton CA, Jones GA, et al. Enhanced pedagogical knowledge and reflective analysis in elementary mathematics teacher education. J Teach Educ. 1996;47:271–282.
[11] Stipec DJ, Givvin KB, Salmon JM, et al. Teachers' beliefs and practices related to mathematics instruction. Teach Teach Educ. 2001;17:213–226.
[12] Smith K. How teachers' beliefs about mathematics affect students' beliefs about mathematics [honors theses. Paper 193]. Durham (NH): University of New Hampshire, University of New Hampshire Scholars' Repository; 2014.