

The Spaghetti Problem

JONAKI GHOSH

The topic of probability forms an important part of the school mathematics curriculum. It is introduced in middle school and is revisited in detail in senior secondary school. Though the concepts of probability can be taught using practical examples, these are usually restricted to tossing of coins, rolling of dice and shuffling playing cards. There are many interesting problems, which can be introduced to enliven the teaching of probability and to enable students to explore the fundamental concepts. In this article we shall explore such a problem called *The Spaghetti Problem* using the spreadsheet MS Excel.

In a previous article titled *The Game of Craps*, which appeared in the November 2019 issue, we explained the importance of simulations as a tool for modeling practical problems. In particular we mentioned *Monte Carlo Simulation*, a technique used to approximate the probability of outcomes of an experiment by running multiple trials generated by random numbers. Simple simulations of real world problems can be explored through spreadsheets such as MS Excel as these are equipped with inbuilt functions for generating random numbers.

The Problem

The problem is fairly simple and is stated as follows:

Let us assume that we are able to randomly break a spaghetti stick of length L into three pieces. What is the probability that the three pieces will form a triangle?

It would be worthwhile to take a spaghetti stick, break it into three pieces and arrange them to form a triangle. This may be repeated a few times using different spaghetti

Keywords: Probability, simulation, spreadsheets

sticks. In some cases the three pieces will form a triangle, while in other cases they will not.

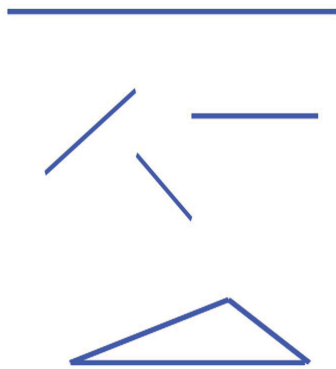


Figure 1. A Spaghetti stick of length L , broken into three pieces, forms a triangle.

In order to estimate the probability, we will need to break a large number of spaghetti sticks, each time checking if the three pieces form a triangle. After a sufficiently large number of trials, we would be in a position to calculate the empirical probability of a triangle being formed. However, it would be tedious and impractical to keep breaking spaghetti sticks. Simulation comes to the rescue here by allowing us to simulate the breaking of sticks using random numbers.

Before describing the process of simulation, let us analyse the problem mathematically.

Consider a stick of length L units, which is broken into three pieces. Let the length of the pieces be x units, y units and $(L - x - y)$ units. The triangle inequality tells us that these *three pieces will form a triangle if the sum of the lengths of any two pieces is greater than the third*. This leads us to the following three inequalities.

$$x + y > L - x - y \quad \text{or} \quad x + y > L/2 \quad (1)$$

$$x + (L - x - y) > y \quad \text{or} \quad y < L/2 \quad (2)$$

$$y + (L - x - y) > x \quad \text{or} \quad x < L/2 \quad (3)$$

Also we observe that both x and y must lie between 0 and L . Let us consider the piece of length x units. If the stick is not broken at all, then x equals L units. If the stick is broken into two pieces, then one will be of length x units and the other will be y units. In this case x is less than L units. The same arguments may be repeated

for the piece of length y units. This leads us to conclude that $0 < x, y < L$. Thus, every time a spaghetti stick is broken into three pieces, an ordered pair of numbers (x, y) is generated which can be easily plotted as a point on the coordinate plane. Also since x and y are greater than 0, all these points will lie in the first quadrant. Now, the inequalities (1), (2) and (3) represent those ordered pairs (x, y) which lead to the formation of a triangle. Graphing the linear equations $x + y = L/2$, $x + y = L$, $x = L/2$ and $y = L/2$ in the first quadrant leads to Figure 2.

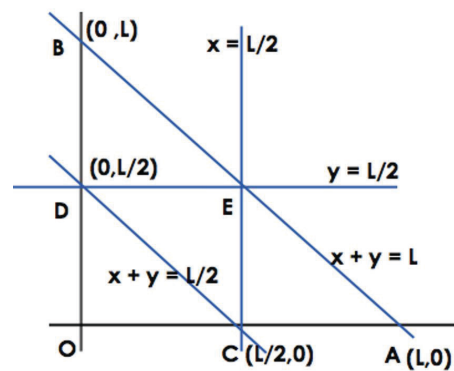


Figure 2: Graphical representation of the linear equations arising out of Spaghetti Problem.

Every time a spaghetti stick is broken, the corresponding ordered pair (x, y) will lie on or within the triangle OAB . This is primarily because of the condition $0 < x, y < L$. Now if we graph the inequalities (1), (2) and (3) simultaneously, the region common to them is represented by the triangle CDE . Hence, if we break a sufficiently large number of sticks such that their corresponding points (x, y) fill up triangle OAB , then the probability of a randomly broken stick (into three pieces) forming a triangle will be given by

Area of triangle CDE / Area of triangle OAB

$$= \frac{\frac{1}{2} \times \frac{L}{2} \times \frac{L}{2}}{\frac{1}{2} \times L \times L} = \frac{\frac{L^2}{8}}{\frac{L^2}{2}} = \frac{1}{4}$$

The above analysis confirms that the *theoretical probability of a spaghetti stick broken randomly, into three parts, forming a triangle is $1/4$* .

The spreadsheet simulation

In order to compute the empirical probability of a triangle being formed out of a randomly broken spaghetti stick, we shall simulate it on MS Excel using its random number generator.

Let us assume that the length of the stick, L , equals 100 units. Suppose that the stick is broken into three parts of lengths a , b and c . We will use the **RANDBETWEEN** command to simulate the lengths a , b and c . Note that **=RANDBETWEEN(1, n)** generates a random integer between 1 and n .

The steps of simulation are as follows.

Step 1: We begin by creating a column of numbers from 1 to 100 in column A. Enter 1 in cell A2 and **= A2 + 1** in cell A3. Take the cursor to the corner of cell A3 and drag till cell A101. This will create a column of numbers from 1 to 100 (as shown in Figure 3) and will help us keep track of the simulations.

	A
1	S.No
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10
12	11
13	12
14	13
15	14
16	15
17	16
18	17
19	18
20	19
21	20

Figure 3: A column of numbers to indicate the number of simulations.

Step 2: We will generate two random numbers between 1 and 100 and call them X and Y. These represent the points at which the stick is broken. For this we enter **=RANDBETWEEN(1, 100)** in cells B2 and C2. The number in cell B2 represents

X and that in C2 represents Y. Selecting cells B2 and C2 simultaneously and double clicking on the corner of cell C2 generates 100 pairs of numbers X and Y.

	A	B	C
1	S.No	X	Y
2	1	29	55
3	2	18	20
4	3	23	46
5	4	37	3
6	5	18	21
7	6	35	46
8	7	70	1
9	8	27	45
10	9	15	17
11	10	27	16
12	11	69	79
13	12	4	86
14	13	99	47
15	14	56	76
16	15	60	37
17	16	35	88
18	17	95	35
19	18	32	64
20	19	23	55
21	20	17	88

Figure 4: The **RANDBETWEEN** command generates two numbers X and Y.

Step 3: The smaller of the two numbers X and Y may be assigned to a . To simulate this we enter the conditional formula

=IF(B2<C2, B2, C2) in D2 (to simulate the length of piece 'a') and double click in the corner of cell D2.

	A	B	C	D
1	S.No	X	Y	a
2	1	29	55	29
3	2	18	20	18
4	3	23	46	23
5	4	37	3	3
6	5	18	21	18
7	6	35	46	35
8	7	70	1	1
9	8	27	45	27
10	9	15	17	15
11	10	27	16	16
12	11	69	79	69
13	12	4	86	4
14	13	99	47	47
15	14	56	76	56
16	15	60	37	37
17	16	35	88	35
18	17	95	35	35
19	18	32	64	32
20	19	23	55	23
21	20	17	88	17

Figure 5: The conditional statement **=IF(B2<C2, B2, C2)** is used to simulate the piece of length 'a'.

Step 4: Once X or Y has been assigned to a, the difference $X - Y$ or $Y - X$ will be assigned to b. This can be achieved as follows

Enter $=IF(B2<C2, C2-B2, B2-C2)$ in E2 (to simulate the length of piece 'b') and double click in the corner of cell E2.

	A	B	C	D	E
1	S.No	X	Y	a	b
2	1	29	55	29	26
3	2	18	20	18	2
4	3	23	46	23	23
5	4	37	3	3	34
6	5	18	21	18	3
7	6	35	46	35	11
8	7	70	1	1	69
9	8	27	45	27	18
10	9	15	17	15	2
11	10	27	16	16	11
12	11	69	79	69	10
13	12	4	86	4	82
14	13	99	47	47	52
15	14	56	76	56	20
16	15	60	37	37	23
17	16	35	88	35	53
18	17	95	35	35	60
19	18	32	64	32	32
20	19	23	55	23	32
21	20	17	88	17	71

Figure 6: The conditional statement $=IF(B2<C2, C2-B2, B2-C2)$ is used to simulate the piece of length 'b'.

Step 5: Finally, to simulate the length of the third piece c, we enter $=100 - (D2 + E2)$ in F2 and double click in the corner of cell F2. Note that each row of the spreadsheet represents one broken spaghetti stick of length 100 units.

	A	B	C	D	E	F
1	S.No	X	Y	a	b	c
2	1	29	55	29	26	45
3	2	18	20	18	2	80
4	3	23	46	23	23	54
5	4	37	3	3	34	63
6	5	18	21	18	3	79
7	6	35	46	35	11	54
8	7	70	1	1	69	30
9	8	27	45	27	18	55
10	9	15	17	15	2	83
11	10	27	16	16	11	73
12	11	69	79	69	10	21
13	12	4	86	4	82	14
14	13	99	47	47	52	1
15	14	56	76	56	20	24
16	15	60	37	37	23	40
17	16	35	88	35	53	12
18	17	95	35	35	60	5
19	18	32	64	32	32	36
20	19	23	55	23	32	45
21	20	17	88	17	71	12

Figure 7: The piece of length 'c' is simulated in column F.

Step 6: We now have a simulation of 100 spaghetti sticks, each broken into three pieces of lengths a, b and c. We need to identify those simulations in which a triangle is formed. The

condition which we shall use is that a triangle will be formed only if the maximum length among a, b and c is less than 50 (half the length of the original stick).

Enter $=IF(MAX(D2,E2,F2)<50, "YES", "NO")$ in G2 (to check if the triangle is formed). The output in cell G2 will be "YES" if a triangle is formed else it will be "NO".

	A	B	C	D	E	F	G
1	S.No	X	Y	a	b	c	Forms a triangle?
2	1	29	55	29	26	45	YES
3	2	18	20	18	2	80	NO
4	3	23	46	23	23	54	NO
5	4	37	3	3	34	63	NO
6	5	18	21	18	3	79	NO
7	6	35	46	35	11	54	NO
8	7	70	1	1	69	30	NO
9	8	27	45	27	18	55	NO
10	9	15	17	15	2	83	NO
11	10	27	16	16	11	73	NO
12	11	69	79	69	10	21	NO
13	12	4	86	4	82	14	NO
14	13	99	47	47	52	1	NO
15	14	56	76	56	20	24	NO
16	15	60	37	37	23	40	YES
17	16	35	88	35	53	12	NO
18	17	95	35	35	60	5	NO
19	18	32	64	32	32	36	YES
20	19	23	55	23	32	45	YES
21	20	17	88	17	71	12	NO

Figure 8: The conditional statement $=IF(MAX(D2,E2,F2)<50, "YES", "NO")$ is used to verify if each row of the simulation represents the formation of a triangle.

Step 7: Finally we need to count the number of cases in which a triangle is formed (indicated by "YES" in column G) to compute the empirical probability. For this we enter $=COUNTIF(G2:G101,"YES")/100$ in any cell, say J2.

The empirical probability = Number of broken spaghetti sticks which form a triangle / total number of spaghetti sticks that are broken

Every time we click on a cell on the spreadsheet, a new set of 100 simulations of broken spaghetti sticks is generated. Each time, the value in cell J2 gives us the empirical probability.

The reader may explore the problem by generating several sets of 100 simulations, each time computing the empirical probability of obtaining a triangle. In some cases the empirical probability may be less than 0.25 and in other cases it may exceed 0.25. It would be worthwhile to generate 1000 simulations (instead of 100) and compute the empirical probability. After

	A	B	C	D	E	F	G	H	I	J
1	S.No	X	Y	a	b	c	Forms a triangle?			
2	1	29	55	29	26	45	YES		Probability	0.26
3	2	18	20	18	2	80	NO			
4	3	23	46	23	23	54	NO			
5	4	37	3	3	34	63	NO			
6	5	18	21	18	3	79	NO			
7	6	35	46	35	11	54	NO			
8	7	70	1	1	69	30	NO			
9	8	27	45	27	18	55	NO			
10	9	15	17	15	2	83	NO			
11	10	27	16	16	11	73	NO			
12	11	69	79	69	10	21	NO			
13	12	4	86	4	82	14	NO			
14	13	99	47	47	52	1	NO			
15	14	56	76	56	20	24	NO			
16	15	60	37	37	23	40	YES			
17	16	35	88	35	53	12	NO			
18	17	95	35	35	60	5	NO			
19	18	32	64	32	32	36	YES			
20	19	23	55	23	32	45	YES			

Figure 9: The empirical probability of a triangle being formed in this set of 100 simulations is equal to 0.26.

several simulations it will be evident that the value of the empirical probability gets closer to the theoretical probability, $\frac{1}{4}$, as the number of simulations increases.

An extension

A natural extension to the problem is to find the probability that a spaghetti stick of length L units, broken into four parts, forms a quadrilateral. We will analyse this situation using a method different from the three pieces case discussed in the previous sections of this article.

Let us break the stick of length L into four pieces such that one piece is of length $L/2$ and the sum of the lengths of the other three pieces is $L/2$. Clearly these four pieces, when joined together, will not form a quadrilateral. In fact a quadrilateral will be formed only when the length of the largest piece is less than the sum of the other three pieces. Thus we need to find the probability that no piece (among the four pieces) has length greater than or equal to $L/2$.

To find the required probability let us consider a circle of circumference L units. (We may imagine that our spaghetti stick is bent into a circle.) Let the four broken parts of the stick be n_1n_2 , n_2n_3 , n_3n_4 and n_4n_1 and let the four points n_1 , n_2 , n_3 and n_4 be randomly placed (in that order) on the circumference of the circle as shown in Figure 10.

Note that the circumference of the circle can be divided into two semi-circles. If the four points n_1 , n_2 , n_3 and n_4 lie on the same semi-circle then the four broken parts will not form a quadrilateral (as in such a case the length of the largest part will be greater than or equal to $L/2$). So, what is the probability of this happening?

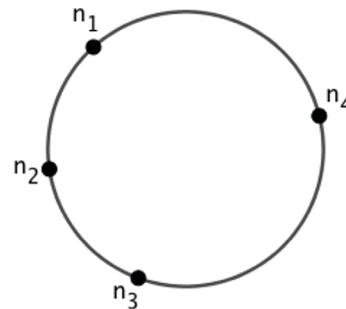


Figure 10. The circle represents a spaghetti stick broken into four parts n_1n_2 , n_2n_3 , n_3n_4 and n_4n_1 .

Consider any one of the four points (say n_1) and the diameter of the circle passing through it, which divides the circumference into two semi-circles (as shown in Figure 11). Now the probability of n_2 lying on either semi-circle is $\frac{1}{2}$ each. Similarly the probabilities of n_3 and n_4 lying on either semi-circle are also $\frac{1}{2}$ each. Thus the probability that all three points, n_2 , n_3 and n_4 lie on the same semi-circle is $\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$. Also, to begin with, we could have chosen any of the four points (n_1 , n_2 , n_3 or n_4) to draw a diameter and thus we have $4C_1 = 4$ choices.

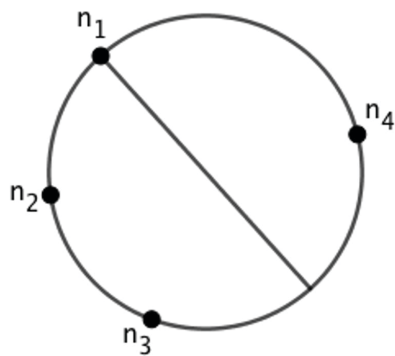


Figure 11. The diameter of the circle passing through n_1 divides the circumference into two semi-circles.

This leads to the conclusion that the probability of all four points lying on the same semi-circle (in which case a quadrilateral cannot be formed) is $4C_1 \times \frac{1}{2^3} = 1/2$.

Hence, *the probability that all four points do not lie on the same semi-circle (that is, a quadrilateral is formed) is $1 - 1/2 = 1/2$.*

The Generalisation

We are now in a position to generalize the spaghetti problem.

If a spaghetti stick is randomly broken into 'n' parts, what is the probability that these parts will form an n-sided polygon?

Clearly (extending from the four parts case discussed in the previous section), the polygon will be formed if the length of the largest part is less than the sum of the remaining $(n - 1)$ parts. Once again, to obtain the solution, we need to find the probability that none of the n parts is of length greater than or equal to $L/2$. We can randomly place the n points on the circumference of a circle (of perimeter L units) as shown in Figure 12.

The reader is urged to prove the following result:

The probability that the n points (shown in Figure 12) lie on the same semicircle is $\frac{nC_1}{2^{n-1}}$.

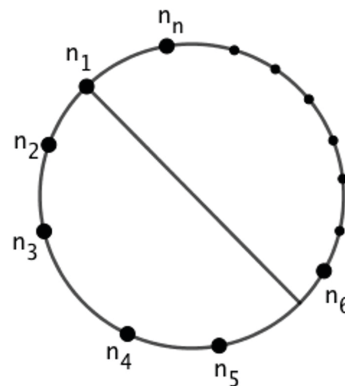


Figure 12. The circle represents a spaghetti stick randomly broken into n parts. The diameter through any one of the n points divides the circumference into two semi-circles.

Hence, the probability that these n points do not lie on the same semi-circle (and that the n broken parts form an n -sided polygon) is $1 - \frac{nC_1}{2^{n-1}}$.

This generalized formula is indeed useful. We can use it to verify the probability of the $n = 3$ case (three spaghetti pieces form a triangle with probability $1/4$) and the probability of the $n = 4$ case (four spaghetti pieces form a quadrilateral with probability $1/2$). The reader is encouraged to work out the probabilities for different values of n . It is interesting to note that as n increases, the probability of the n pieces forming an n -sided polygon also increases!

An interesting follow-up activity is to design a simulation of the $n = 4$ case on MS Excel and verify that the empirical probability of a quadrilateral being formed approaches the theoretical probability $1/2$ as the number of trials increases. Students of secondary and senior secondary school can easily explore this problem. Students of grades 9 and 10 will be able to appreciate the initial analysis of the $n = 3$ case using the triangle inequality and also the spreadsheet simulation. Students of grades 11 and 12 with a stronger grounding in probability, are likely to be curious about the $n = 4$ case and the generalization of the problem. The discussion of the general case is well within the reach of most students.

Integrating Computational thinking and Mathematical Thinking

In recent years computational thinking (CT) has been identified as one of the key analytical abilities required for mathematics and science learning. The rapidly changing nature of scientific and mathematical disciplines and the need to prepare students for careers in these disciplines have been the primary motivation for bringing computational thinking into classroom practices. While many definitions of CT may be found in the literature, Seymour Papert [5,6,7] was the first to emphasise the importance of computational thinking and its connection to mathematics learning. Papert vividly talked about children using computers as instruments for learning and for enhancing creativity, innovation, and was actually responsible for "concretizing" the term computational thinking. With turtle geometry and logo programming he introduced a computational style of learning geometry. Later, in 2006, the term computational thinking was greatly popularized by Jeanette Wing [9] in her seminal article in which she advocated that

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic (the three R's), we should add computational thinking to every child's analytical ability.

She further elaborated on what encompasses computational thinking. According to her

Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science.

Weintrop et.al. [8] attempt to define computational thinking in the context of school mathematics and science education and also suggest a theoretical grounding for the same. They propose a taxonomy comprising

four categories: data practices, modeling and simulation practices, computational problem solving practices and systems thinking practices. All definitions or frameworks that define CT emphasise a certain set of skills. These include the ability to deal with challenging problems, representing ideas in computationally meaningful ways, creating abstractions for the problem at hand, breaking down problems into simpler ones, assessing the strengths and weaknesses of a representation system and engaging in multiple paths of inquiry. These skills are also critical for mathematics learning and there is a common consensus on the understanding that CT skills have to be developed in mathematics classrooms right from the early school years. It is also evident that among all school subjects, mathematics can provide ample contexts to integrate CT and computation and can in turn enrich mathematics learning through technology. Mathematical thinking (MT) skills and Computational thinking (CT) practices are distinct and yet mutually supportive.

However, operationalizing CT in mathematics classrooms is a pedagogical challenge. It requires identifying tasks that are both mathematically and computationally rich – tasks that allow students to explore concepts, develop and employ CT practices and also foster MT. Digital tools such as dynamic software, computer algebra systems and spreadsheets can play a significant role in mediating CT and MT. In fact spreadsheets can be very powerful tools for CT-MT based explorations as they can aid in identifying patterns, generating graphical representations and creating simulations. They are very suited for the inquiry based approach to learning and do not require high level coding skills. The spaghetti problem and other similar problems, which connect to various topics in the mathematics curriculum and are amenable to exploration via simulations, can be easily modeled on spreadsheets. The Spaghetti problem exploration may be divided into three stages. We may identify the role of MT and CT in each of these stages.

1. *Visualising the problem:* Applying the triangle inequality to model the problem and arriving at the inequalities (1) to (3) requires MT whereas modifying the problem for simulating it on the spreadsheet entails CT.
2. *Solving the problem:* Graphing the inequalities (1) to (3), arriving at the feasible region and computing the theoretical probability requires MT while simulating the problem on the spreadsheet (using random number generators and conditional statements) to obtain the empirical probability entails CT.
3. *Generalising the problem:* Extending the problem to 4 or n pieces and calculating the theoretical probability entails MT whereas extending the spreadsheet simulation to larger number of pieces requires CT.

We are now in a position to claim that exploration of the Spaghetti problem and other similar problems can help to foster both CT and MT. The reader is encouraged to read the articles *The Birthday Paradox* [3] and *The Monty Hall Problem* [4], which appeared in earlier issues of the magazine, to deliberate on possibilities offered by these problems for integration of CT and MT in the mathematics classroom.

Bibliography

- [1] Basic Excel Formulas retrieved from <https://corporatenanceinstitute.com/resources/excel/study/basic-excel-formulas-beginners/>
- [2] Exploring mathematics using spreadsheets: The Spaghetti Problem retrieved from <https://www.youtube.com/watch?v=1mCS632FQmM>
- [3] Ghosh, J. B. (2012). The Birthday Paradox in *At Right Angles*, Azim Premji University, Vol. 1 (1), 39 – 42.
- [4] Ghosh, J. B. (2012). The Monty Hall Problem in *At Right Angles*, Azim Premji University, Vol. 1 (2), 57 – 62.
- [5] Papert, S. (1972). Teaching children to be mathematicians versus teaching about mathematics. *International Journal of Mathematical Education in Science and Technology* 3(3), 249–262.
- [6] Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic books, New York.
- [7] Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematics Learning* 1(1), 138–142.
- [8] Weintrop, D., Beheshti, E., Horn, M., Orton, k., Jona, K., Trouille, L. & Wilensky, U. (2016). *Defining Computational Thinking for Mathematics and Science Classrooms*, *Journal of Science Education and Technology* 25, 127–147.
- [9] Wing J.M. (2006). Computational thinking. *Commun ACS*. 49(3), 33 – 35.



JONAKI GHOSH is an Assistant Professor in the Dept. of Elementary Education, Lady Sri Ram College, University of Delhi where she teaches courses related to math education. She obtained her PhD in Applied Mathematics from Jamia Milia Islamia University, New Delhi, and her M.Sc. from IIT Kanpur. She has taught mathematics at the Delhi Public School, R K Puram, where she set up the Math Laboratory & Technology Centre. She has started a Foundation through which she conducts professional development programmes for math teachers. Her primary area of research interest is in the use of technology in mathematics instruction. She is a member of the Indo Swedish Working Group on Mathematics Education. She regularly participates in national and international conferences. She has published articles in journals and authored books for school students. She may be contacted at jonakibghosh@gmail.com.