Edwards, M. T. (2006). Shutting the box: Fostering collaboration among early grades and secondary preservice teachers through authentic problem solving. C*ontemporary Issues in Technology and Teacher Education,* 6(4), 374-398.

"Shutting the Box": Fostering Collaboration Among Early Grades and Secondary Preservice Teachers Through Authentic Problem Solving

Michael Todd Edwards
Miami University (Ohio)

Abstract

In this paper are discussed recent efforts to provide preservice mathematics teachers with opportunities to connect elementary teaching methods and content with the content and methods of secondary school mathematics. Through an in-depth exploration of the game, Shut the Box, preservice elementary and secondary mathematics teachers thoughtfully analyzed and manipulated computer-generated output, developed and tested their own conjectures, and collaboratively answered questions involving theoretical probabilities *across courses and content levels*. Through their collaboration, the preservice teachers gained a better appreciation of mathematics content and pedagogical strategies that lie beyond the grades they will likely teach, as they reconsidered the importance of content and pedagogical knowledge at every level of mathematics instruction. These interactions are considered in this document through a discussion of the mathematical underpinnings of the popular board game.

Results of curricular and classroom investigations of the Third International Mathematics and Science Study (TIMSS, 1995) reported that time in the American mathematics classroom is more likely spent learning specific concepts and skills—at a somewhat shallow level—without providing the deep connections that experts believe will "improve students' ability to learn and understand a subject in an integrated way" (National Center for Educational Statistics, 1997).

This state of affairs lies in direct contrast to recommendations set forth in the Curriculum Principle and the Connections Standard of the *Principles and Standards for School Mathematics* (National Council Teachers of Mathematics [NCTM], 2000). Two major goals for American mathematics classrooms detail the need for a coherent curriculum and classroom experiences that help students develop a connected, integrated, holistic view of mathematics, rather than experience it from year to year, or grade to grade, as a disconnected discipline (NCTM, 2000).

Against this backdrop of the K-12 mathematics classrooms, I sought a preventive approach that might build on a specific recommendation for teacher preparation. It highlights the need for prospective teachers to make connections extending beyond their specific grade-level boundaries (Conference Board of the Mathematical Sciences [CBMS], 2001). Perhaps successful classroom teachers—particularly those with significant, varied classroom teaching experience—ably and meaningfully communicate connections among various mathematical concepts and levels of mathematics. Unfortunately, prospective mathematics teachers (hereafter referred to as preservice teachers or PSTs) often struggle to make these same connections for their students. As a result many PSTs do not have a good sense of where their students "came from" or where they "are going" mathematically. They may, therefore, benefit from experiences in pedagogy and content courses that prepare them to make mathematical connections beyond *their particular level* of certification (or licensure).

In this article, is described an approach and the resulting benefits in which various technology-based mathematics and communication tools (e.g., programming and data analysis utilities, computer algebra systems, flowcharting software, and Internet-based research applications) were used to explore connections between elementary- and secondary-level mathematics. While building new mathematical knowledge related to probability elementary and secondary mathematics, PSTs collaboratively engaged in authentic mathematical activity: generating original conjectures, testing hypotheses, and taking "wrong turns."

The elementary mathematics PSTs need college mathematics experiences in which *their* ideas for solving problems are elicited and taken seriously, their sound reasoning affirmed, and their missteps challenged in ways that help them make sense of their errors (CBMS, 2001, p. 17). To further enrich their mathematical understanding they also need to explore and develop connections to the mathematical concepts that extend beyond the grades for which they will be licensed (CBMS, 2001).

Similarly, secondary mathematics PSTs need knowledge of the mathematical concepts and skills that students acquire before they reach the high school years (CBMS, 2001). When provided with opportunities to collaborate—particularly when engaged in authentic, long-term problem-solving projects—both groups (elementary and secondary mathematics PSTs) may begin to break the "cycle of disconnect" that exists between their respective groups and become better prepared for teaching mathematics to their prospective students.

Recent efforts to provide mathematics PSTs with opportunities to connect elementary teaching methods and content with the methods and content of secondary school mathematics are shared in this article. Through an in-depth exploration of the popular board game, Shut the Box (Red Fern Enterprises, 2006), elementary and secondary PSTs thoughtfully analyzed and manipulated computer-generated output, developed and tested their own conjectures, and collaboratively answered questions involving theoretical probabilities *across courses and content levels*.

Through their collaboration, the PSTs gained a better appreciation of mathematics content and pedagogical strategies that lie beyond the grades they will likely teach. They reconsidered the importance of content and pedagogical knowledge at many levels of mathematics instruction and, as hoped, they experienced "their own capacity for mathematical thought" (in the words of CBMS, 2001, p. 24). These interactions are considered in the context of discussions of the mathematical underpinnings of Shut the Box.

Basics of the Game

According to legend, the game originated in northern France more than 200 years ago as a recreation for fishermen and sailors (Masters, 2006). In typical versions of the game, players use standard dice and a playing tray that features a row of tiles numbered 1-9. Each tile on the playing tray is moveable. Typically, tiles may be "flipped over" or "slid" by means of a hinge or sliding cover. Figure 1 illustrates two variations of the traditional gameboard.

Figure 1. *Examples of typical Shut the Box gameboards. Permission to use images provided by Red Fern Enterprises (left image) and Independent Living Aids (right image).*

Players take turns rolling dice. After each roll, a player "flips" one or two tiles with a sum equal to the value of the dice roll. Once tiles are flipped over, players cannot flip them again. In other words, no tile may be flipped more than once. When the sum of the unflipped tiles is less than or equal to 6, a single die is rolled. A round ends when tiles can no longer be flipped. A score for a particular round is the sum of the numbers on unflipped tiles. Lower scores are more desirable. In fact, the ultimate goal of each round is to flip over all nine tiles, at which point players "shut the box." Figure 2 illustrates a typical round, along with possible moves after each dice roll.

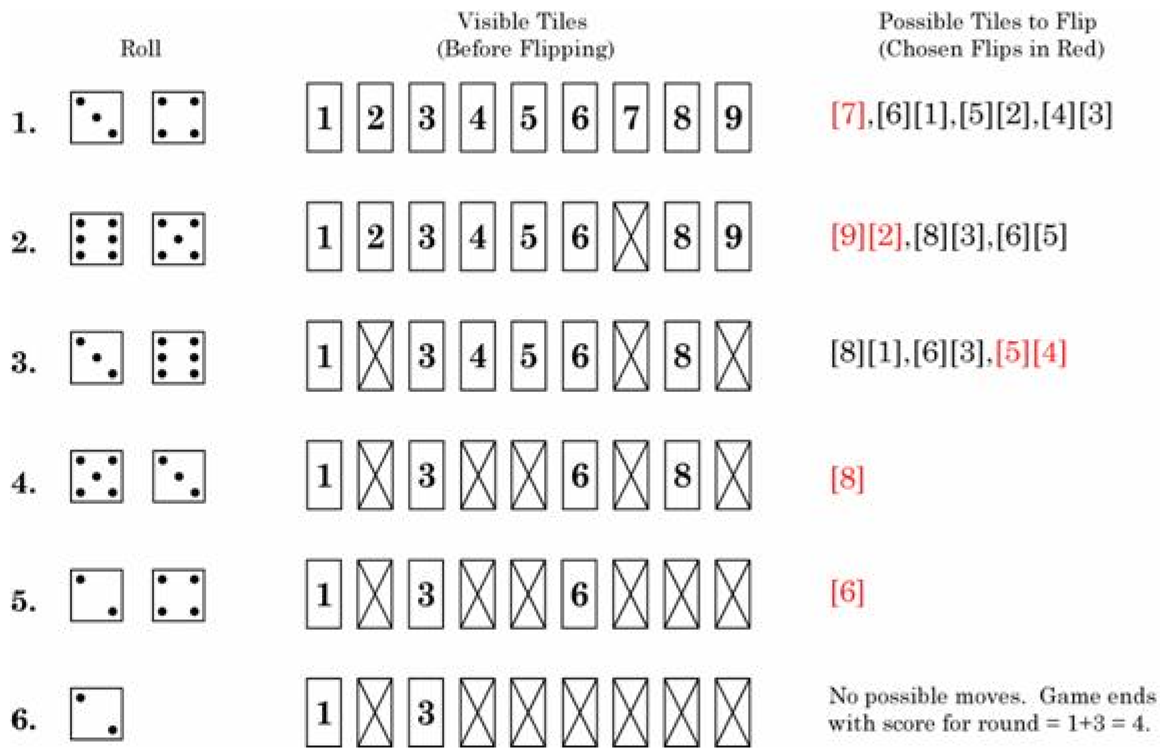


Figure 2. *Typical game of Shut the Box.*

The game is interesting (and applicable) for both elementary and secondary mathematics teachers. Several features of the game make it appealing and mathematically relevant to a wide audience.

- In the lower grades, the game may be used to motivate practice with addition and subtraction of whole numbers. Students practice addition facts as they calculate dice rolls, final scores, and possible tile flips.
- In the middle grades, the game may be used to explore fundamental notions of probability.
- In the secondary grades, the game may be used to motivate the study of concepts such as independence and probabilities of compound events.
- For all grade levels, the study of the distribution of final scores of various rounds of game-play provides students with opportunities to engage in meaningful data collection and analysis tasks.

With these observations in mind, the game was presented to PSTs in two very different courses: Adolescent Education Special Methods (ED 337), a mathematics methods course offered to secondary mathematics PSTs, and Mathematics for Elementary School Teachers (MT 171), a mathematics content course offered to elementary PSTs. As students in both courses analyzed, they engaged in discussions of mathematics and pedagogy across traditional course boundaries. Several unexpected instructional twists and turns associated with the analysis of the game led to genuinely meaningful collaboration among the groups.

Instructional Design Phases

In the remainder of this article, let us consider the PSTs' investigation of Shut the Box in four distinct phases: (a) Introduction to Shut the Box; (b) Preliminary Investigation of Probabilities; (c) Internet-based Research; and (d) Verification of Findings. Typically the two classes worked in parallel, with findings and artifacts constructed in one class informing the work of the other. The level of teacher interaction across courses varied throughout the investigation. On one occasion, elementary PSTs presented materials and content to secondary PSTs. In numerous instances, they shared ideas asynchronously (e.g., simulation code, diagrams, and articles) via our university's *Blackboard*™ courseware system. The parallel configuration of the courses, phases of the study, and the interaction between the PSTs are illustrated in Figure 3 and clarified further in this article.

Introduction to Shut the Box

Initially, Shut the Box was introduced to elementary and secondary PSTs using technology-oriented tools with slightly different emphases based on the mathematical relevance of the game to each class of PSTs and instructional aims of each course.

Introduction to ED 337 students. Because ED 337 is a teaching methods course, attention was initially focused on methods for capturing the interest of secondary students (i.e., instructional "hooks"). To this end, the secondary PSTs were shown video clips of a popular 1980s game show, *High Rollers* (available at the time of this study at www.youtube.com). The rules of High Rollers are similar to the rules of *Shut the Box*, with the exception that players always use two dice when playing High Rollers (whereas in our variation of Shut the Box, one die is rolled when the sum of the visible tiles is less than 6).

In the *High Rollers* clip viewed in class, the contestant shuts the box in a dramatic fashion, rolling two 1s to clear all numbers from the game board. The clip was used to motivate several probability-related questions in the ED 337 classroom: (a) What was the probability of the contestant rolling a "2" on the last roll? and (b) What was the probability of any contestant actually "shutting the box" during a typical round of High Rollers? What about during a typical game of Shut the Box?

After a discussion of these probabilities (note that the answer to the second question is far from trivial) and several rounds of play with a Web-based version of the game (Garsha, 2006), the secondary PSTs were provided with a Shut the Box worksheet intended for middle grades mathematics students (Glencoe/McGraw-Hill, 2006, Lesson 11-5). Working in pairs, they identified state academic content standards addressed by the activity and completed other related tasks for homework.

ED 337 homework. The following Shut the Box tasks were assigned for homework: (a) Modify and expand the teaching ideas provided within the Glencoe/McGraw Hill worksheet to develop a mathematics lesson plan suitable for use with high school students. Identify several state academic content standards addressed by the modified lesson plan; and (b) use any available tools (including but not limited to mathematical proof, computer programming, or computer algebra related techniques) to determine the theoretical probability of shutting the box with all tiles (i.e., those numbered 1-9) initially visible.

Introduction to MT 171 Students. MT 171 is a content-centered course offered to prospective elementary mathematics teachers. Typically, the elementary PSTs enrolled in MT 171 have taken less mathematics coursework than their secondary counterparts (who must earn a major in mathematics in order to obtain a teaching license in our state). For these reasons, when Shut the Box was introduced to elementary PSTs in this course, more emphasis was placed on the mathematics content of the game. For instance, after playing several rounds of Shut the Box, the elementary PSTs considered the mathematical understanding required for young children to play the game meaningfully.

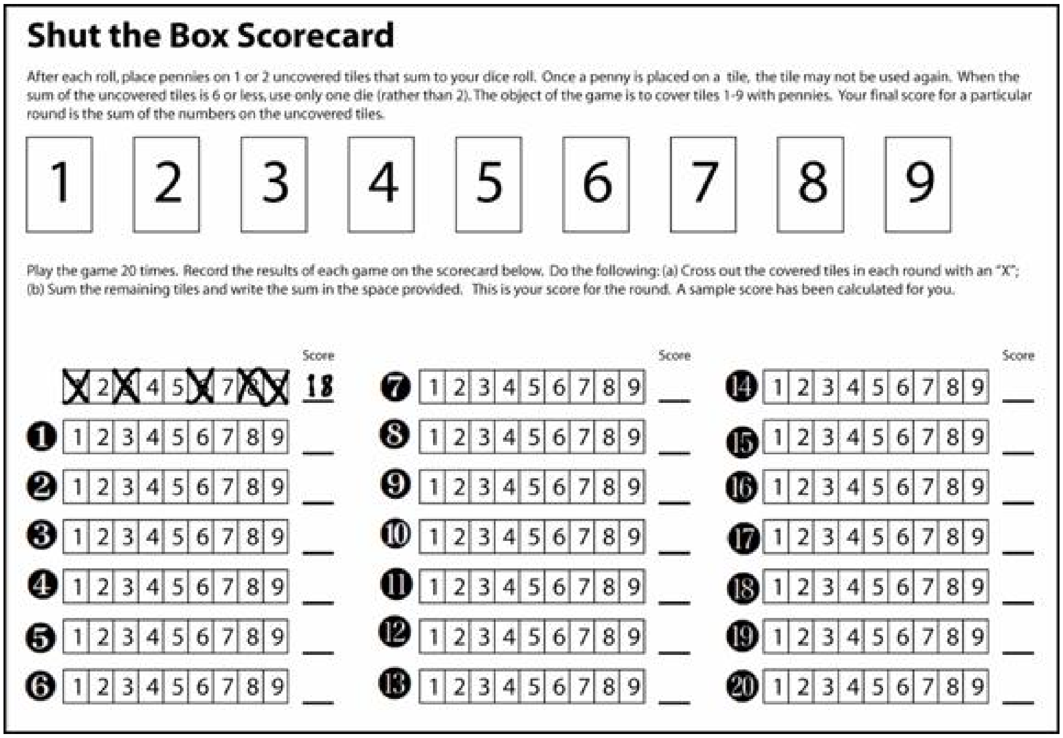


Figure 4. *Shut the Box data collection sheet.*

MT 171 homework. In contrast to the work assigned to secondary PSTs, the tasks for elementary PSTs were wholly content oriented. Furthermore, given the mathematical backgrounds of the elementary PSTs, the Shut the Box tasks purposefully included more scaffolding as they were asked to complete the following tasks:

1. Using a Shut the Box gameboard, dice, and counters, play 20 rounds of the game, and keep track of final scores for each round (the results of this task were ultimately used to compare theoretical and experimental probabilities). Figure 4 shows a sample scorecard that the MT 171 teachers used to record Shut the Box score data.
2. Determine the theoretical probability of "shutting the box" under the following conditions:

   a. With *one* dice roll, with 5 the only remaining visible tile.

   b. With 1 and 3 the only remaining visible tiles.

c. With tiles 1, 3, and 4 the only remaining visible tiles.

d. With all tiles initially visible.

Note that task 2(d) was assigned to both groups of PSTs. Ultimately, the decision to provide scaffolding for the homework of the elementary PSTs was fortuitous, because their solutions provided both groups (ED 337 and MT 171) with initial ideas for solving the probability problem.

Preliminary Investigation of Theoretical Probabilities

Both groups of PSTs experienced difficulty when calculating the theoretical probability of shutting the box with all tiles initially visible. Their approaches were influenced by previous coursework and the differing nature of the tasks assigned to each group.

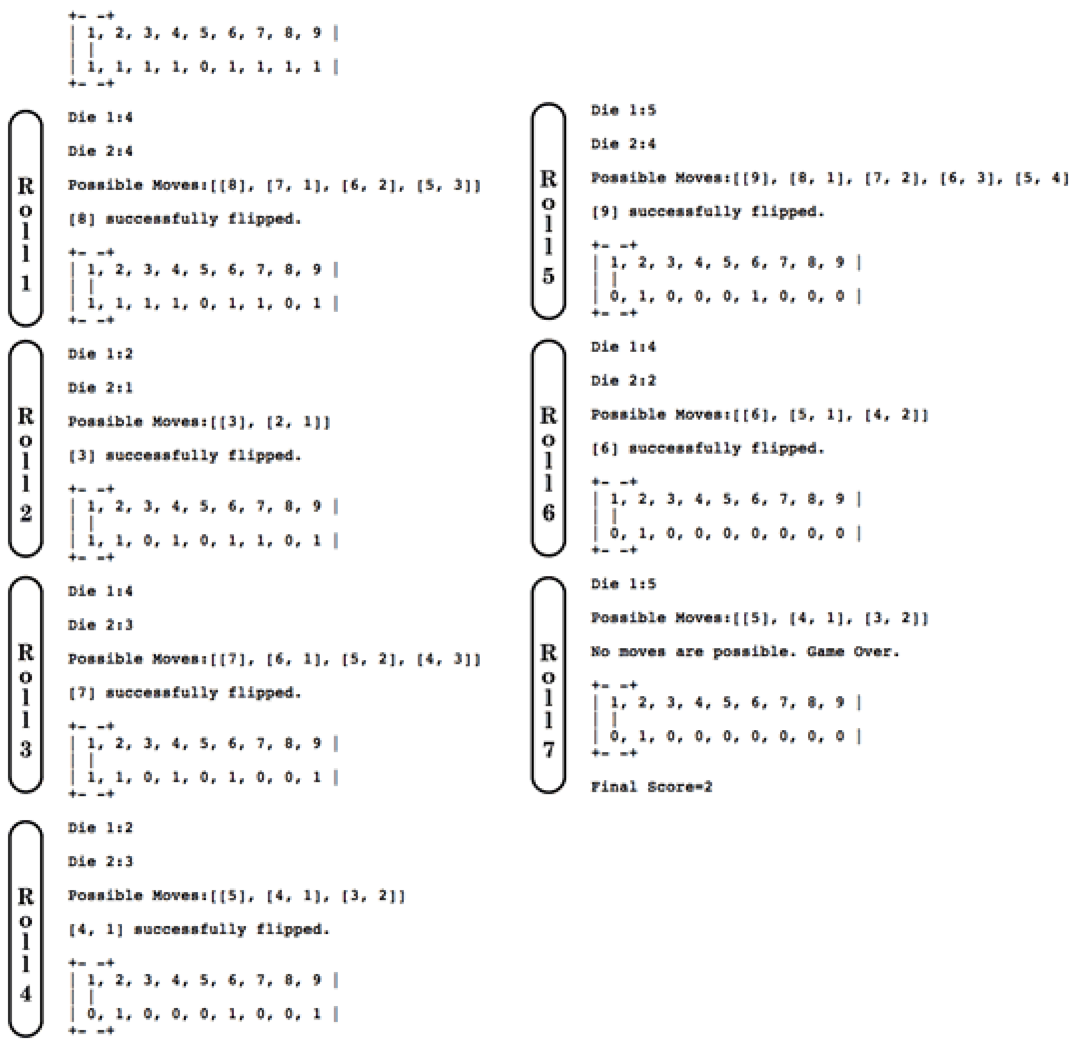


Figure 5. *Sample output generated by* `ShutBox_Sim_Verbose` *simulation.*

Initial work of secondary PSTs. Prior to their enrollment in ED 337, the secondary mathematics PSTs successfully completed two computer-related courses: Introduction to Computer Science (CS 201) and a co-requisite computer lab course (CS 201L). As such, they had a strong knowledge of basic computer programming concepts, including variables, assignments, conditionals, loops, procedures, functions, and parameter passing. Moreover, they were experienced Computer Algebra System (CAS) users since most used such software extensively during three semesters as calculus students.

A group of four ED 337 PSTs applied their knowledge of CAS and programming to construct two (SciFace Software GmbH & Co., 2006) programs that simulate Shut the Box game-play while calculating experimental probabilities. The first program, `ShutBox_Sim_Verbose`, simulates the step-by-step game-play of a single trial of Shut the Box. The program uses two `random()` function calls to simulate die rolls and utilizes a 9x2 matrix to represent the Shut the Box gameboard. A revised version of the `ShutBox_Sim_Verbose` code is provided in Appendix A.

Annotated output of the program, shown in Figure 5, allows one to trace the simulated game-play. As the output suggests, the simulation successfully implements the "one die rule" when the sum of visible tiles is less than or equal to 6. Furthermore, the code successfully calculates the sum of the remaining tiles at the end of the game.

The second program, `ShutBox_Sim_Mult_Trials`, repeatedly executes trials of Shut the Box and stores the resulting score of each trial into an array. After all trials are completed, a bar graph of various final scores and the ratio of "box shutting" trials to overall trials (i.e., the "experimental" probability) are output. Sample output from the program is shown in Figure 6. The code itself is similar to the `ShutBox_Sim_Verbose` code and is provided in Appendix B.

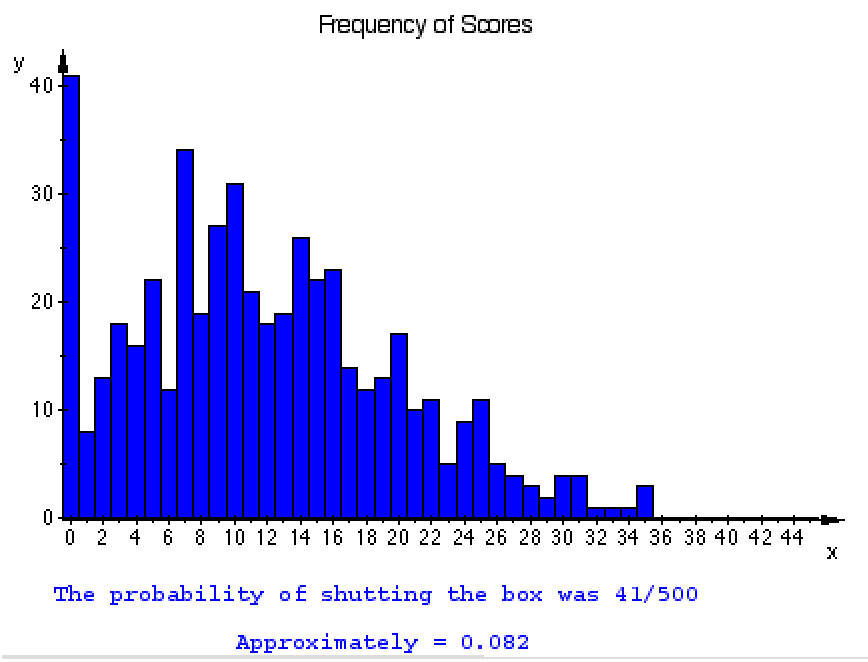


Figure 6. *Sample output generated by* `ShutBox_Sim_Mult_Trials` *simulation.*

Although the code-based solutions constructed by the secondary PSTs in ED 337 did not adequately answer the question initially posed (recall that they were asked to calculate *theoretical* probabilities rather than experimental ones) their initial work enhanced the instruction I provided to the elementary PSTs.

- The code was useful for highlighting differences between experimental and theoretical probability.
- Final scores generated by MuPAD provided teachers with a context for exploring various data displays and measures of central tendency.
- The programs helped launch discussions of the "law of large numbers" and suitable sample size.

Initial work of elementary PSTs. The solutions to the Shut the Box problems provided by elementary PSTs differed markedly from those generated by the secondary PSTs. Differences are most likely explained by the scaffolded nature of the problems given to the elementary PSTs, as well as the content of the MT 171 course.

Particularly interesting was the tendency of the elementary PSTs to calculate probabilities using tree diagrams. Having studied tree diagrams prior to the introduction of the Shut the Box game, several PSTs attempted to answer the Shut the Box probability problems using them. Figure 7 illustrates a strategy for calculating the theoretical probability of shutting the box with tiles 1 and 3 remaining. In the diagram, dice rolls are represented as numbers within parentheses; for example, (4) represents the die roll of 4. Visible tiles are represented as numbers within brackets, for example [3] represents the 3 tile.

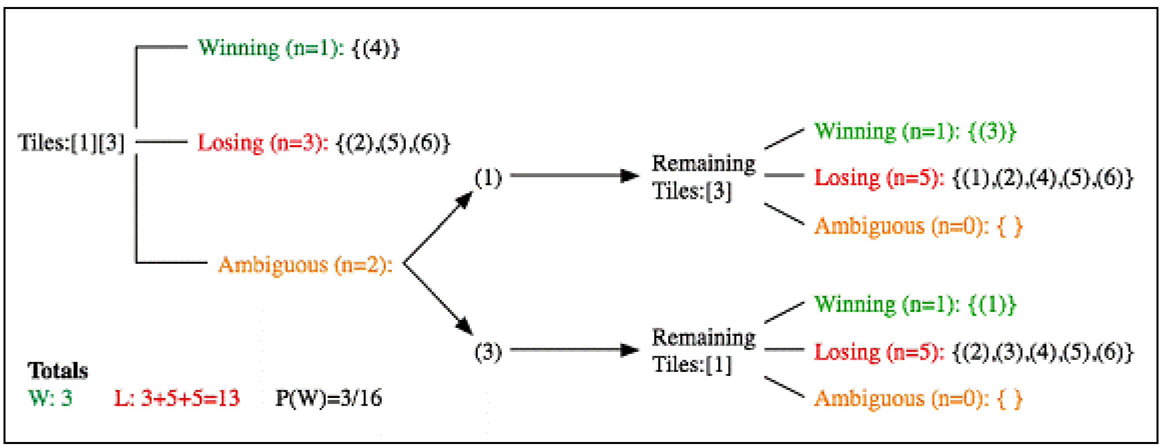


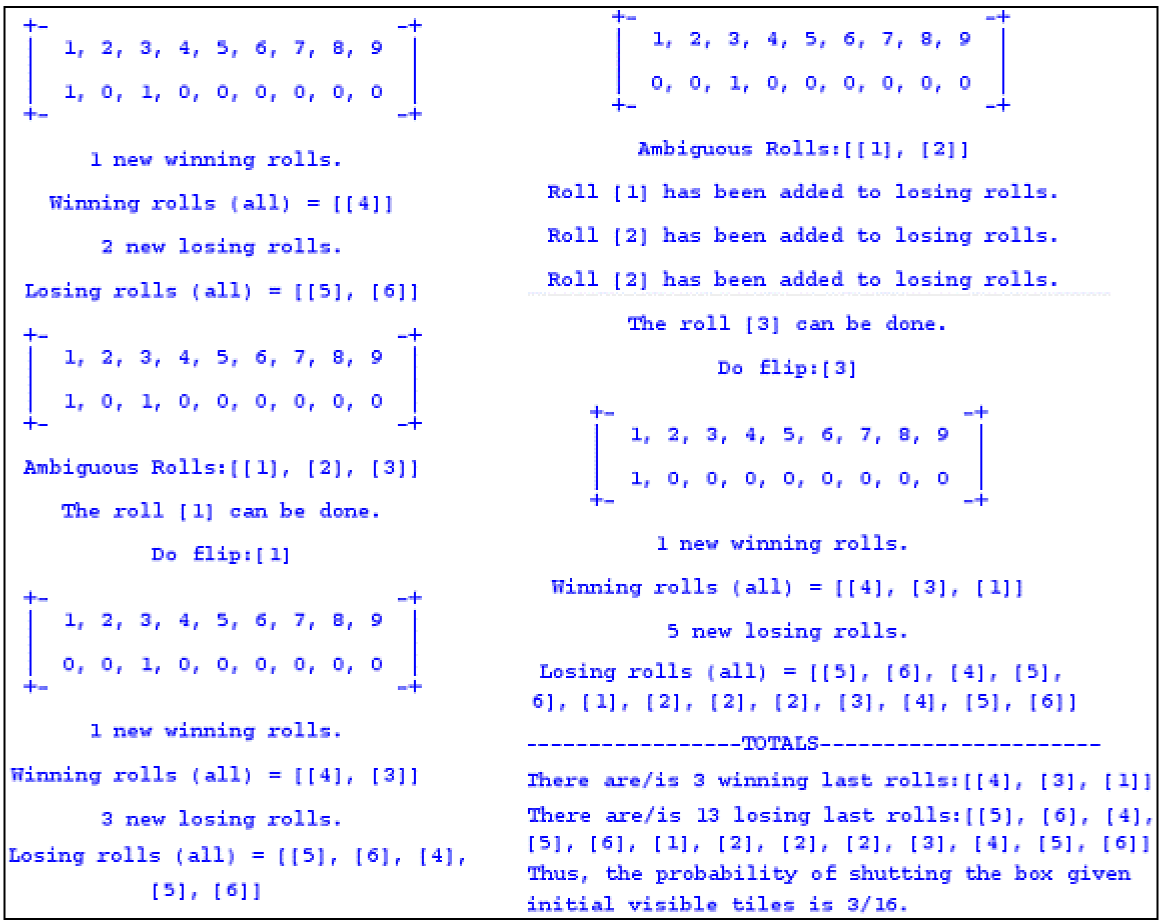
Figure 7. *Tree diagram for calculating theoretical probability of shutting box with tiles 1 and 3 initially visible.*

The initial state of the game is depicted at the left side of the tree diagram with all possible die rolls from this state that is, (1), (2), (3), (4), (5), and (6) — placed to the immediate right. Each roll is placed in one of three possible categories: winning, losing, and ambiguous. A winning die roll results in the box being shut. A losing die roll ends the game with unflipped tiles. An ambiguous die roll fails to end the game and requires further analysis.

The tree diagram indicates three distinct ways of shutting the box with tiles 1 and 3 initially visible, namely with die rolls (4), (1), (3), and (3), (1). On the other hand, the chart indicates 13 ways to lose the game. Using the tree diagram, teachers calculated the probability of shutting the box as 3/16.

Several elementary PSTs were assigned the task of constructing tree diagrams with OmniGraffle (Omni Group, 2006), a popular charting software. (Note: Demonstration versions of the software are downloadable at http://www.omnigroup.com/applications/omnigraffle/.) The diagrams were subsequently uploaded for public download in the Course Documents sections of both the ED 336 and the MT 171 Blackboard courseware sites.

Tree-diagramming code. Although calculating theoretical probabilities with tree diagrams can be cumbersome if more than two or three tiles are initially visible, the secondary PSTs were impressed by the approach. In particular, they were intrigued by the idea of categorizing rolls as winning, losing, or ambiguous and saw possibilities for automating the technique through computer programming. In fact, one group of secondary PSTs successfully implemented the tree diagramming strategy depicted in Figure 7 in two related MuPAD programs — `ShutBox_Theory_Verbose` and `Shutbox_Theory_Terse`. Both programs calculate theoretical probabilities by categorizing rolls as winning, losing, or ambiguous. Sample output from the verbose program is provided in Figure 8 for a game in which tiles 1 and 3 are initially visible.


```
+-                          -+
|  1, 2, 3, 4, 5, 6, 7, 8, 9 |
|  1, 0, 1, 0, 0, 0, 0, 0, 0 |
+-                          -+

1 new winning rolls.
Winning rolls (all) = [[4]]
2 new losing rolls.
Losing rolls (all) = [[5], [6]]

+-                          -+
|  1, 2, 3, 4, 5, 6, 7, 8, 9 |
|  1, 0, 1, 0, 0, 0, 0, 0, 0 |
+-                          -+

Ambiguous Rolls:[[1], [2], [3]]
The roll [1] can be done.
Do flip:[1]

+-                          -+
|  1, 2, 3, 4, 5, 6, 7, 8, 9 |
|  0, 0, 1, 0, 0, 0, 0, 0, 0 |
+-                          -+

1 new winning rolls.
Winning rolls (all) = [[4], [3]]
3 new losing rolls.
Losing rolls (all) = [[5], [6], [4],
[5], [6]]

+-                          -+
|  1, 2, 3, 4, 5, 6, 7, 8, 9 |
|  0, 0, 1, 0, 0, 0, 0, 0, 0 |
+-                          -+

Ambiguous Rolls:[[1], [2]]
Roll [1] has been added to losing rolls.
Roll [2] has been added to losing rolls.
Roll [2] has been added to losing rolls.
The roll [3] can be done.
Do flip:[3]

+-                          -+
|  1, 2, 3, 4, 5, 6, 7, 8, 9 |
|  1, 0, 0, 0, 0, 0, 0, 0, 0 |
+-                          -+

1 new winning rolls.
Winning rolls (all) = [[4], [3], [1]]
5 new losing rolls.
Losing rolls (all) = [[5], [6], [4], [5],
6], [1], [2], [2], [2], [3], [4], [5], [6]]
-----------------TOTALS----------------------
There are/is 3 winning last rolls:[[4], [3], [1]]
There are/is 13 losing last rolls:[[5], [6], [4],
[5], [6], [1], [2], [2], [2], [3], [4], [5], [6]]
Thus, the probability of shutting the box given
initial visible tiles is 3/16.
```

Figure 8. *Output generated by* `ShutBox_Theory_Verbose` *code.*

The `ShutBox_Theory_Terse` code is similar to the verbose code; however, its output is limited to lists of final winning and losing rolls and a final theoretical probability. The output is identical to the totals section of the verbose code shown in Figure 8. Note that in Figure 8, the theoretical probability generated by the verbose program matches that calculated by elementary PSTs using tree diagramming methods. This was consistently the case.

Unfortunately, when either the verbose or terse code is executed with more than three or four tiles initially visible, run times are quite long. For instance, when attempting to calculate the theoretical probability of shutting the box with tiles 1-9 initially visible, the program executed continuously for more than 3 *days*, using more than 300 MB of machine memory (using an Apple Powerbook with a 1.67 GHz Power PC G4 processor and 512 MB of SDRAM), until execution was finally halted. The brute force nature of the tree diagramming algorithm, despite its usefulness, was simply too inefficient to solve large problems.

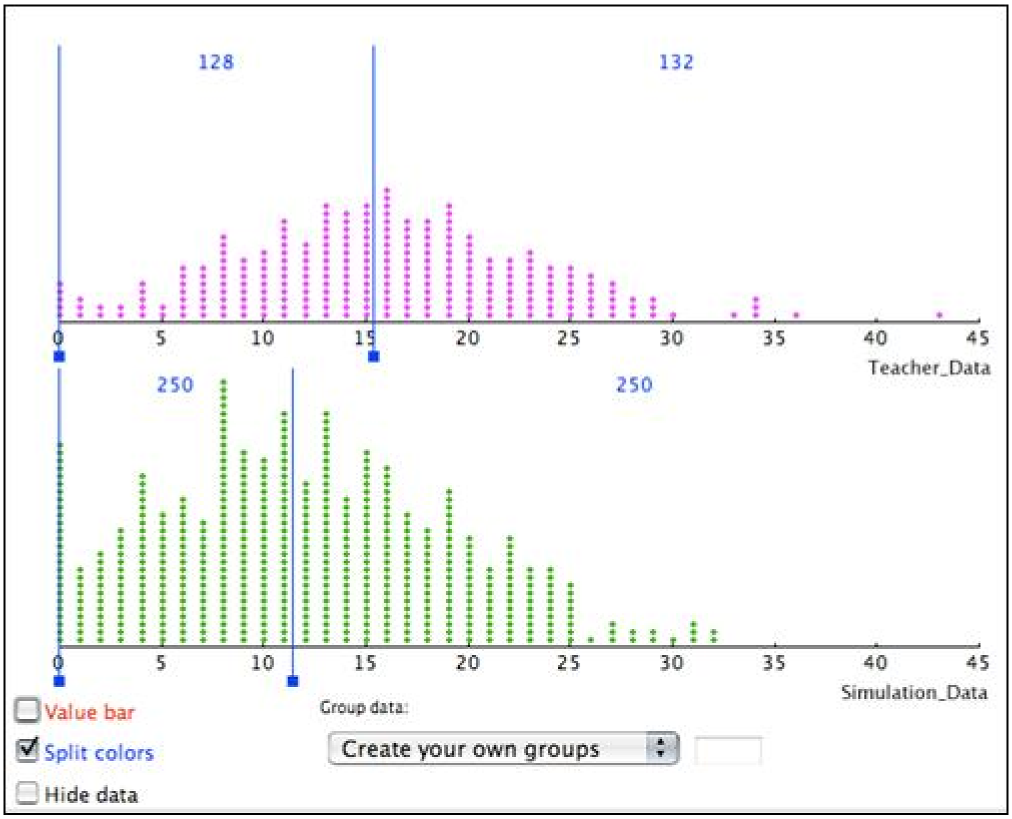


Figure 9. *Dot plots of simulated and teacher data as generated by an NCTM Navigations (Bright, 2003) java applet by Koeno Gravemeijer, Paul Cobb, and Kay McClain © 1998 Vanderbilt University.*

Comparing simulation results to teacher data. Simulation data generated by the secondary PSTs' `ShutBox_Sim_Mult_Trials` code (refer to Appendix B) was compared with aggregate Shut the Box data compiled by 13 elementary PSTs. Simulation data and aggregate data were stored into a text file and analyzed using a java applet provided with the *NCTM Navigating Through Data Analysis in Grades 6-8* text (Bright, 2003). A screenshot of the applet, provided in Figure 9, highlights dot plots of two datasets: one consisting of scores from 500 simulated rounds of Shut the Box, and the other consisting of scores generated by the 13 elementary PSTs (*n* = 260).

As the plots in Figure 9 suggest, the simulation code appears to generate scores that, on average, are smaller than PST-generated scores. In particular, the box was shut more frequently by the simulation than by the elementary PSTs using actual dice. This finding aroused the curiosity of both groups of PSTs. Were the apparent differences attributable to faulty programming? Or was the computer choosing tile flips more strategically? Was there a way to determine best moves for the game for particular dice rolls? Without knowing the theoretical probabilities associated with ending the game, it was difficult (if not impossible) to answer such questions.

At this point, curiosity regarding Shut the Box was heightened both by incomplete results provided by brute force methods and by apparent discrepancies between PST- and simulation-generated scores. We continued exploration beyond the amount of time initially allocated for the activity. Ultimately, through Internet-based research and collaboration, the PSTs were able to answer many of the previously generated questions while uncovering significant errors in their previous work.

Internet-Based Research

Numerous Web sites are devoted to the game-play or history of Shut the Box, yet relatively few discuss the mathematical underpinnings of the game. Two noteworthy exceptions include the Web sites Durango Bill's "Shut the Box" Analysis (Butler, 2006) and An Analysis of the Game Shut the Box (Hunt, 2003). Both sites discuss optimal strategies for shutting the box and probabilities associated with shutting the box for particular board configurations. In particular, both make reference to recursive strategies for calculating theoretical probabilities.

Hunt (2003) provided C++ source code that implements a recursive approach. However, the PSTs—even those with considerable programming experience—were unable to explain significant portions of the code. Hunt (2003) noted that it the code "was knocked up in an afternoon and it seems to work. That is the only claim I am prepared to make about it. There are probably better ways of writing this program" (Hunt, 2003).

Butler (2006), on the other hand, clearly described the recursive process for calculating probabilities (as "pseudocode"), but does not provide actual computer code on his site. Benjamin and Stanford (1995) described specifics for calculating Shut the Box probabilities as an example of dynamic programming. Like Butler, however, they did not provide source code.

In the absence of well-documented code to determine Shut the Box probabilities, the secondary PSTs converted the recursive pseudocode of Benjamin, Stanford, and Butler into a working C++ program. The code, which comprised the bulk of the PSTs' ED 337 final project, is described in detail in the following section. Aspects of the continued collaboration between secondary and elementary PSTs are also further elaborated.

Coding Specifics

Facets of the recursive pseudocode of Benjamin and Stanford (1995) and Butler (2006) relevant to both elementary and secondary PSTs are highlighted as follows.

Binary numbers and Shut the Box. Butler (2006) described a method for representing all possible gameboard configurations that is clearly connected to the content of a typical "mathematics for elementary school teachers" course.

For ease of coding, Butler (and others) considered the tiles of a Shut the Box gameboard to be arranged in *descending order from left to right* (in reverse order of the tiles on the Shut the Box gameboard; see Figure 10).

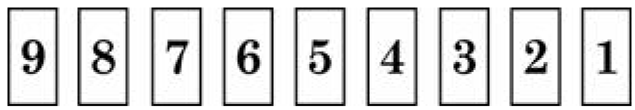


Figure 10. *For ease of coding, the Shut the Box gameboard is considered in reverse order.*

Such a configuration mimics the conventional Hindu-Arabic numbering system more closely, with the most significant digits occurring in the left-most positions.

Butler's (2006) pseudocode shows he recognized that each game tile must be in exactly one of two possible states, flipped or unflipped. In the pseudocode, a placeholder is assigned to keep track of the state of each tile (see Figure 11). When a tile is flipped, a 1 is stored in the tile's placeholder. When a tile is unflipped, a 0 is stored in the tile's placeholder. For instance, 001000101 represents the configuration with tiles 1, 3, and 7 flipped and all others unflipped as depicted in Figure 11.

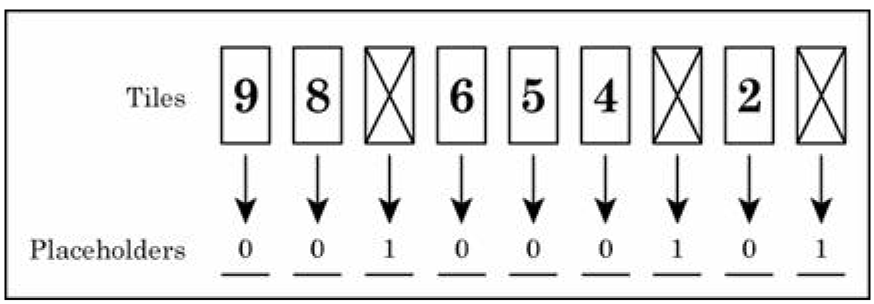


Figure 11. *Placeholders with tiles 1, 3, and 7 flipped.*

Viewed as a binary (base 2) value, the number 001000101 equals

$$0\cdot 2^8 + 0\cdot 2^7 + 1\cdot 2^6 + 0\cdot 2^5 + 0\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 = 64 + 4 + 1 = 69.$$

In a similar manner, *every* configuration of flipped and unflipped tiles can be represented by a binary number ranging from 000000000 (no tiles flipped) to 111111111 (all tiles flipped). Converted to base 10, these numbers range from 0 to 511 and represent exactly 512 different configurations of flipped and unflipped tiles for the Shut the Box game.

Surprisingly enough, representing an integer in a different base (such as base 2) is an unfamiliar topic for most secondary PSTs. On the other hand, alternate number bases are a foundational topic for elementary PSTs studying basic integer operations. Because the PSTs wishing to implement the recursive pseudocode needed to understand binary numbers, several elementary PSTs agreed to construct an alternate number base tutorial (i.e., a guided mini-lesson) for them and discuss the lesson during one of the ED 337 class meetings. While contributing to conversations across these groups, the lesson construction and teaching tasks were graded as part of the presentation component for the elementary PSTs in the MT 171 course.

Recursive problem solving. Benjamin and Stanford (1995) described dynamic programming in the following terms.

> Dynamic programming can be thought of as the art of working backwards. We find optimal strategies to problems by building on already-computed optimal strategies to "smaller" problems. Usually, we need to specify the optimal solutions to only the simplest (often trivial) problems, together with a recurrence that builds upon solutions to smaller problems to create solutions to larger ones. (p. 12)

The dynamic programming strategies discussed by Butler (2006), Benjamin and Stanford (1995), and Hunt (2004) suggested that students calculate the probabilities of easy board configurations first, then use those probabilities to calculate the probabilities of more difficult board configurations. This strategy is both powerful and applicable to a wide variety of problems.

Recognizing that it is easier to calculate the probability of shutting the box when more tiles are flipped, the PSTs initiated a recursive problem-solving method by sorting the various board configurations from easiest to most difficult. Butler (2006) recommended using the sum of unflipped tiles as a measure of the relative difficulty of a particular board configuration. This measure agrees with the PSTs' previous tree diagramming experiences, since calculating theoretical probabilities was far easier when all but one or two tiles were already flipped.

Calculating probabilities. To determine the probability of shutting the box from a particular board configuration, one calculates the probability of shutting the box from the configuration for every possible dice sum. Since any two dice sums are *mutually exclusive* (e.g., you can not roll a 3 and a 5 simultaneously), the probabilities associated with each roll are added to calculate an overall probability. Furthermore, to calculate the probability of shutting the box from a particular board configuration for a specific dice sum, one should multiply the probability of rolling that specific sum by the probability of shutting the box after tiles have been flipped.

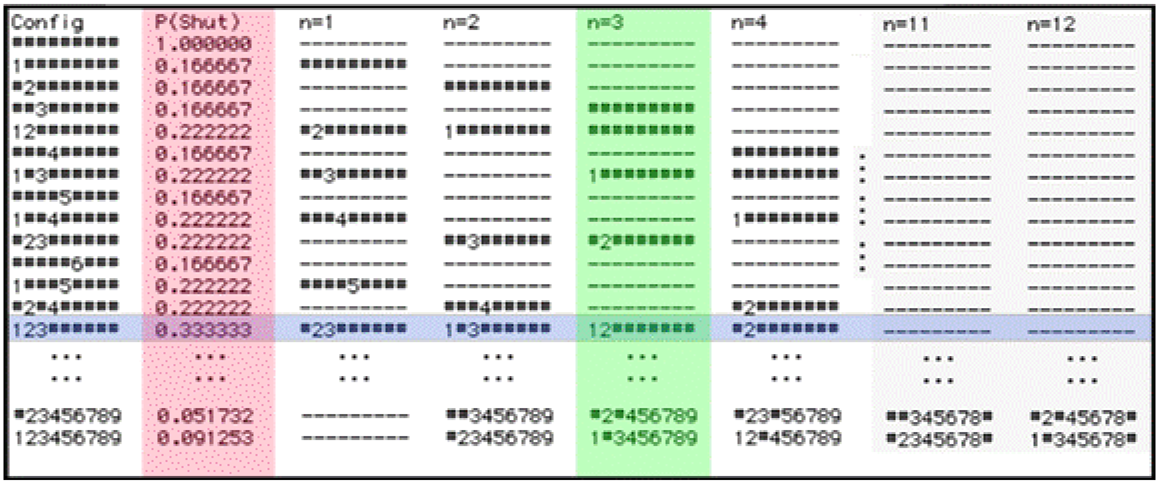
To make the details of these calculations more concrete, the elementary and secondary PSTs calculated the probability of shutting the box for the four simplest board configurations (boards 511, 510, 509, and 508, respectively) by hand. The calculations, similar to those discussed by Butler (2006), are provided in tabular format (Appendix C).

The most easily obtained probability (i.e., that associated with shutting the box from configuration 511) was used to calculate the probabilities for configurations 508, 509, and 510. In this way, the examples (Appendix C) illustrate the recursive nature of dynamic programming.

After this initial introduction, elementary PSTs applied this technique to calculate the probability of shutting the box for the next 10 configurations and stored the results in a spreadsheet. The results were shared via *Blackboard* and used by the secondary PSTs as they tested dynamic programming code.

Verification of Findings

Ultimately, the secondary PSTs successfully translated the pseudocode into a working program (`shutbox_dynamic.cpp`) that calculated theoretical probabilities for "shutting the box" for each of the 512 possible board configurations. Unlike the MuPAD trees code, which literally consumed *hours* of processing time, the C++ code terminated successfully in *less than 5 seconds*. A portion of the output is provided in Figure 12 (the entire output, along with the C++ source code appears in Appendix D and Appendix E.



Config	P(Shut)	n=1	n=2	n=3	n=4	n=11	n=12
#########	1.000000	---------	---------	---------	---------	---------	---------
1########	0.166667	#########	---------	---------	---------	---------	---------
#2#######	0.166667	---------	#########	---------	---------	---------	---------
##3######	0.166667	---------	---------	#########	---------	---------	---------
12#######	0.222222	#2#######	1########	#########	---------	---------	---------
###4#####	0.166667	---------	---------	---------	#########	---------	---------
1#3######	0.222222	##3######	---------	1########	#########	---------	---------
####5####	0.166667	---------	---------	---------	---------	---------	---------
1##4#####	0.222222	###4#####	---------	---------	1########	---------	---------
#23######	0.222222	---------	##3######	#2#######	---------	---------	---------
#####6###	0.166667	---------	---------	---------	---------	---------	---------
1###5####	0.222222	####5####	---------	---------	---------	---------	---------
#2#4#####	0.222222	---------	###4#####	---------	#2#######	---------	---------
123######	0.333333	#23######	1#3######	12#######	#2#######	---------	---------
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
#23456789	0.051732	---------	##3456789	#2#456789	#23#56789	##345678#	#2#45678#
123456789	0.091253	---------	#23456789	1#3456789	12#456789	#2345678#	1#345678#

Figure 12. *Partial output generated from* `shutbox_dynamic.cpp` *code.*

Interpreting the output. Every possible board configuration, with easiest ones appearing first, is listed in the leftmost (Config) column of the program output. Individual rows of the output contain information regarding particular configurations. For example, information pertaining to configuration `123######` (i.e., tiles 1, 2, and 3 initially visible) is listed in the `123######` row (shaded light blue in Figure 12).

Probabilities of shutting the box for various configurations are listed on the corresponding rows in the `P(Shut)` column (shaded pink). For example, the probability of shutting the box for configuration `123######` is 0.333333 or 1/3. The "*n* = 3" column (shaded green in Figure 12) lists optimal next flips for various configurations when a 3 is rolled. For instance, if a player rolls a 3 from the `123######` configuration, the probability of shutting the box by flipping tile 3 (i.e., by transforming `123######` into `12#######`) changes the probability from 1/3 to 2/9 (shown as 0.222222).

For the PSTs, it was particularly interesting to note that the C++ program reports that when all tiles are initially unflipped, a player can expect to shut the box slightly more than 9% of the time (0.091253) This value, typically larger than the probabilities reported by the PSTs' simulation code (refer to Figure 6), aroused concerns regarding the efficiency of the simulation code.

A closer inspection of the `shutbox_dynamic.cpp` output was simultaneously encouraging and troubling. Although the probabilities of shutting the box for various configurations matched the hand calculations performed by the PSTs' (see Appendix C), the probabilities did not match earlier results derived from tree diagramming. For instance, the `shutbox_dynamic.cpp` code reports that the probability of shutting the box from configuration 1#3###### is 2/9 (or 0.222222), whereas the tree diagrams of the PSTs (refer to Figure 7) indicated a probability of 3/16 (or 0.1875).

For more than an hour of class time, secondary PSTs in the ED 337 course attempted (in vain) to determine why the probabilities generated from dynamic programming conflicted with those generated from tree diagramming. Ultimately, a pair of elementary PSTs identified the source of the conflicting results. Namely, their initial tree diagrams overlooked differences in probabilities associated with specific "wins." This oversight also explained differences in data generated by the secondary PSTs' simulation code and actual dice rolls of the elementary PSTs illustrated in Figure 9). For instance, when tiles 1 and 3 are initially visible, the rolls (4) and (1,3) both shut the box. However, theoretically the probability of rolling a 1 followed by a 3 is 1/36, while the theoretical probability of rolling a 4 is only 1/6. When tree diagrams were revised to take these differences into account, the two approaches (simulation code and tree diagramming) generated the same probabilities (see Figure 13).

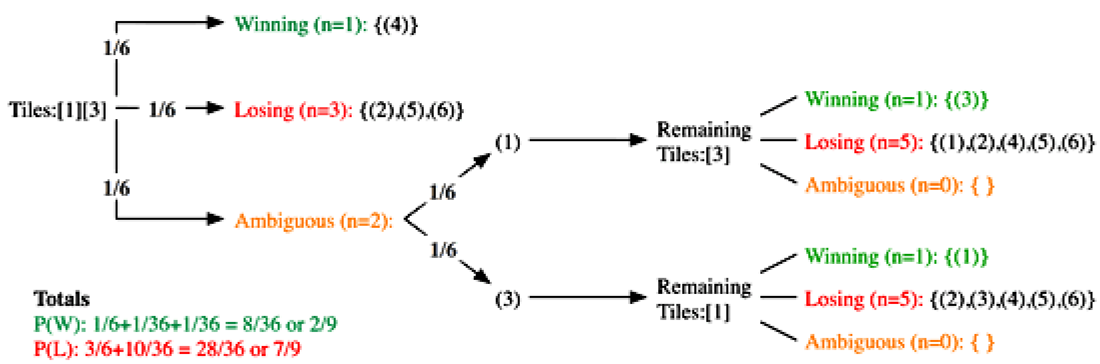


Figure 13. *Revised "tree diagram" for configuration* 1#3######.

Conclusions and Implications for Future Research

Although the extent of collaboration (primarily sharing files and code asynchronously) among the elementary and secondary PSTs during the study of Shut the Box was modest, the benefits of the shared problem exploration were evident. In the following paragraphs, general reactions of the PSTs are shared and ideas for future study are provided.

Reactions of Elementary and Secondary PSTs

The experience of investigating a single "big problem" for multiple weeks (approximately 4 weeks) was a new experience for most, if not all, of the PSTs. The collaboration across traditional course and content boundaries, however modest, was equally unfamiliar. Uncharted experiences presented the PSTs with challenges and opportunities. Although some PSTs found the open-ended nature of the problem investigation unsettling (comments such as “Where is class heading?” “Will we ever finish this problem?” and “Do you know the answers to these questions?” were not uncommon), others cited benefits of the exploration. Several of the more positive benefits cited may be summarized with these three themes: (a) a reconsideration of teacher myths; (b) a recognition of mathematics as a connected discipline; and (c) a better understanding of authentic mathematics instruction.

A reconsideration of teacher myths. The interaction among the PSTs encouraged them to reconsider myths and stereotypes of those preparing to teach at other levels. For instance, at the end of the study, several secondary PSTs expressed genuine surprise regarding the problem-solving skills of the elementary PSTs (e.g., "I was surprised that the elementary teachers were able to see things that I couldn't," "I didn't think [elementary teachers] liked math or knew much about it"). Likewise, at the end of the study, several elementary PSTs noted that they discovered that "high school teachers don't know everything"—a revelation for many. During my office hours, one elementary PST commented that the experience represented the "first time ever someone valued the way I solve math problems."

A recognition of mathematics as a connected discipline. The investigation of Shut the Box encouraged PSTs to see connections between elementary and secondary mathematics. The game itself appeals to a wide audience, with various aspects of game-play equally well-suited for study by students in the elementary and secondary grades. In several instances, the mathematical content knowledge of one group was instrumental in the problem-solving efforts of the other group. For instance, the elementary PSTs' understanding of binary numbers and tree diagrams aided secondary PSTs as they calculated probabilities. Through classroom discussions of the game, secondary PSTs learned about the emphasis on invented algorithms in the elementary years. Each PST level gained a better understanding of the important role of recursion in the problem solving process.

A better understanding of authentic mathematics instruction. The experience of exploring unsolved problems collaboratively represented an authentic learning experience for the PSTs. The importance of such activity is noted in the Curriculum Principle of the *Principles and Standards of School Mathematics* (NCTM, 2000).

> The curriculum also must focus on important mathematics—mathematics that is worth the time and attention of students ... Topics such as recursion, iteration, and the comparison of algorithms have emerged and deserve increased attention because of their relevance. (p. 15)

Through their exploration of Shut the Box—in particular, through a comparison of tree diagramming and dynamic programming solutions—teachers experienced the relevance of recursion, iteration, and the comparison of algorithms. In a sense, the entire experience represented an "iterative process" since teachers generated conjectures then constructed and tested possible solutions at each stage of the investigation.

For many of the PSTs, engagement in mathematical study of this sort was new. When the answer to a particular question was unknown or when they had to rely on one another rather than the "back of the book" or the "tutoring center," the PSTs felt uncomfortable. As such, engaging PSTs in activities such as Shut the Box has the potential to impact future practice. Open-ended, authentic mathematical tasks (such as the Shut the Box project) provide teachers with opportunities to experience ways in which a series of problems can be differentiated and stretched to meet learners' needs at many levels. Here some PSTs constructed different mathematical products (e.g., hand-drawn tree diagrams, computer programs, written explanations of mathematics concepts), used different mathematical processes to solve problems (e.g., software, pencil-and-paper calculations), and explored different mathematical content (e.g., theoretical probability, experimental probability, binary numbers, recursion) as they explored significant mathematical topics. Such experiences provide new teachers with models for differentiating instruction in their own lesson planning and in their own practice.

Future Steps

The benefits of even modest levels of collaboration suggest that increased collaboration among content and methods courses, beyond that explored in this study, would likely benefit PSTs. Although the use of technology aided in this collaboration, tools such as wikis or videoconferencing, would certainly expand the collaboration beyond asynchronous interaction. I plan to develop more problems conducive to collaboration and explore them in a more collaborative fashion. Presenting problems and their solutions with wikis opens up the possibility of including practicing teachers more easily in discussions of content and pedagogy. A collection of wikis discussing big problems and their solutions would serve as a worthwhile resource for practicing and preservice teachers as well as teacher educators.

At the conclusion of the project, I recognized the enormous potential of such projects for encouraging teacher reflection with respect to the use of technology to support/encourage differentiated instruction. As PSTs solve authentic, inquiry-based mathematical problems, they should be encouraged to reflect upon the implementation of similar tasks in their own classrooms. For instance, as teachers collaboratively solve open-ended mathematics tasks, they may be asked to construct lesson plans that provide similar experiences for their own students (although perhaps, on a smaller scale initially). Shut the Box provides teachers with opportunities to envision how they can use technology s an inquiry tool in their own instruction, and how even the “least expert” students can contribute to collaborative problem solving efforts in meaningful ways.

References

Benjamin, A. & Stanford, D. (1995). Optimal Klappenspiel. *The UMAP Journal 16*(1), 11-20. Available online at http://www.math.hmc.edu/~benjamin/papers/Klappenspiel.pdf

Bright, G. (2003). *Navigating through data analysis in Grades 6–8* (CD ROM). Reston, VA: National Council of Teachers of Mathematics.

Butler, B. (2006). *Durango Bill's shut the box analysis*. Retrieved April 20, 2006, from http://www.durangobill.com/ShutTheBox.html

Conference Board of the Mathematical Sciences. (2001). *The mathematical education of teachers: Part I*. Washington, DC: Mathematical Association of America.

Garsha, L. (2006). *ShutBox* (Java Applet). Retrieved March 27, 2006, from http://www.angelfire.com/games2/ShutBox/sb.html

Glencoe/McGraw-Hill (2006). *Mathematics: Applications and concepts - Course 1*. Columbus, OH: Author.

Hunt, T. (2003). *An analysis of the game shut the box*. Retrieved April 20, 2006, from http://www.timhunt.me.uk/shut_the_box/index.html

Independent Living Aids, Inc. (2006). *Accessible math games.* Retrieved May 25, 2006, from http://www.independentliving.com/departments.asp?dept=177 &deptname=Math.

Masters, James (2006). *The online guide to traditional games: Shut the box - History and useful information*. Retrieved May 25, 2006, from http://www.tradgames.org.uk/games/Shut-The-Box.htm

National Center for Education Statistics. (1997). *Pursuing excellence: A study of U.S. eighth-grade mathematics and science teaching, learning, curriculum, and achievement in international context*. Retrieved September 15, 2006, from http://www.nces.ed.gov/pubs97/timss/97198-6.asp

National Council of Teachers of Mathematics. (2000). *Principles and standards for school mathematics*. Reston, VA: Author.

Omni Group. (2006). *OmniGraffle Pro* (Version 4.1.1). Seattle, WA: Author. (Demonstration versions available for download at http://omnigroup.com/applications/omnigraffle/)

Red Fern Enterprises, Inc. (2006). *Shut the box*. Retrieved May 25, 2006, from http://www.backporchgames.com/shut-the-box.html

SciFace Software GmbH & Co. (2004). *MuPAD* (Version 3.1). Patterborn, Germany: Author. (Demonstration versions available for download at http://www.mupad.com)

Authors Note:

Michael Todd Edwards
Miami University (Ohio)
m.todd.edwards@gmail.com

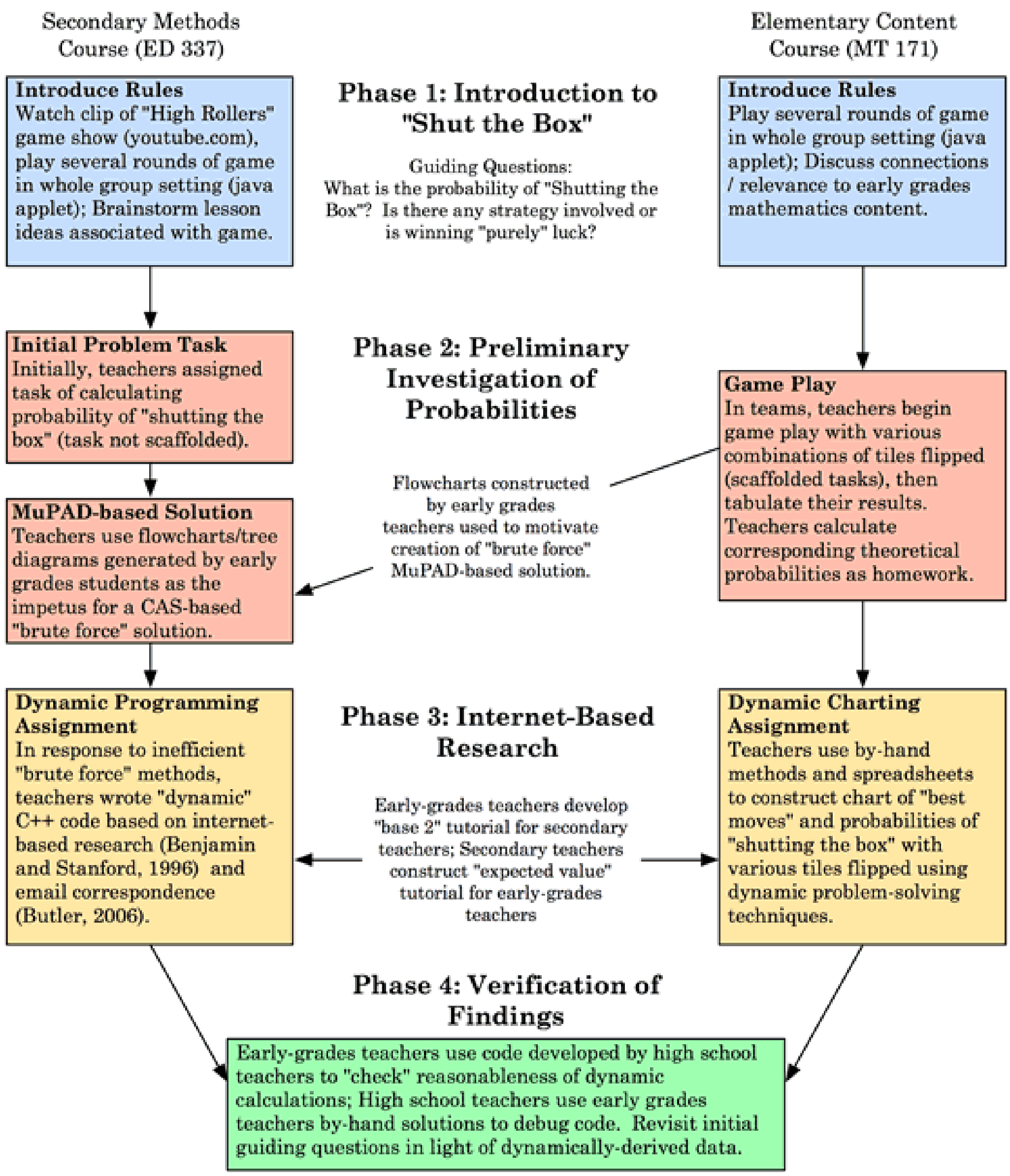


Figure 3. *Instructional phases of Shut the Box activities.*

Appendix A

`ShutBox_Sim_Verbose` Code (*MuPAD* 3.1).

The code simulates the game play of one trial of *Shut the Box*.

```
// ShutBox_Sim_Verbose - ED 337, April 12, 2006
// This code simulates a single game of "Shut the Box".
diceRoll1:=random(1..6):diceRoll2:=random(1..6):
theTiles := matrix([[1,1], [2,1],[3,1],[4,1],[5,1],[6,1],[7,1],[8,1],[9,1]]):
// theTiles is a 9 row x 2 column matrix representing tiles on the "shut the
// box" game board.  The first column contains the numerical label of each tile
// (i.e. 1-9).  The second column contains a "0" or a "1". An entry of 0 next
// to a particular number indicates that the corresponding tile has been
// "flipped."  When the second column of theTiles contains all 0's, this
// indicates that the "box has been shut".
print(linalg::transpose(theTiles)): //Print initial gameboard
move_is_possible:=TRUE:
while (move_is_possible <> FALSE) do
   theVisibleSum:=0;
   for k from 1 to 9 do // Calculate the sum of all unflipped tiles.
        theVisibleSum:=theVisibleSum+theTiles[k,1]*theTiles[k,2]:
   end;
   die1:=diceRoll1():theSum:=die1:print(Unquoted,"Die 1:".expr2text(die1));
   if (theVisibleSum > 6) then
      die2:=diceRoll2():theSum:=theSum+die2:
      print(Unquoted,"Die 2:".expr2text(die2));
   end: //moves is a list of possible combos of tile flips
   moves:=combinat::partitions::list(theSum,MaxPart=9,MaxSlope=-1,MaxLength=2):
   print(Unquoted,"Possible Moves:".expr2text(moves)):
   number_of_possible_moves:=nops(moves):
   tiles_flipped:=FALSE:
   for i from 1 to number_of_possible_moves do //the code cycles through all
   //possible moves (based on dice sum) until one that can be made is found
        flips:=op(moves,i):      // flips=ith possible move (a list)
        flips_available:=1:      // Set flips_available = TRUE (i.e. 1)
        number_of_flips:=nops(flips):      // The number of integers in the list
        for j from 1 to number_of_flips do
           theFlip:=op(flips,j):      // examine integers individually
           flips_available:=flips_available*(theTiles[theFlip,2]):
        end;
        if (flips_available=1) then  //if 1, all flips were "available"
           for j from 1 to number_of_flips do
                theFlip:=op(flips,j):theTiles[theFlip,2]:=0:
           end: //make newly flipped numbers"unavailable" (i.e. 0)
           print(Unquoted,expr2text(flips)." successfully flipped."):
           tiles_flipped:=TRUE:print(linalg::transpose(theTiles)):break:
        end:
        if ((tiles_flipped=FALSE) and (i=number_of_possible_moves)) then
           print(Unquoted,"No moves are possible.  Game Over."):
           print(linalg::transpose(theTiles)):
           theVisibleSum:=0;
           for k from 1 to 9 do  // Calculate the sum of all unflipped tiles.
                theVisibleSum:=theVisibleSum+theTiles[k,1]*theTiles[k,2]:
           end;
           print(Unquoted,"Final Score=".expr2text(theVisibleSum)):
           move_is_possible:=FALSE:
        end: // end if statement
   end: // end for statement
end: // end while
```

Appendix B

ShutBox_Sim_Mult_Trials Code

```
// ShutBox_Mult_Trials - ED 337, April 16, 2006
// This code simulates multiple trials of the game "Shut the Box".
numtrials:=500: // numtrials is the number of trials of the game
theScores := [0 $ i = 0..45]: //theScores is an array that contains the number
  // of times each possible score (i.e. 0-45) is generated by multiple trials
  // of "Shut the Box."  In general, theScores[k+1] holds the number of times
  // the score k is generated (note that no "0th" element exists in a list,
  // so indices are "off by 1."  For instance, theScores[1] holds the number of
  // times the box was "shut".
for n from 1 to numtrials do
  diceRoll1:=random(1..6):diceRoll2:=random(1..6):
  theTiles := matrix([[1,1], [2,1],[3,1],[4,1],[5,1],[6,1],[7,1],[8,1],[9,1]]):
  move_is_possible:=TRUE:
  while (move_is_possible <> FALSE) do
      theVisibleSum:=0;
      for k from 1 to 9 do // Calculate the sum of all unflipped tiles.
          theVisibleSum:=theVisibleSum+theTiles[k,1]*theTiles[k,2]:
      end;
      die1:=diceRoll1():theSum:=die1:
      if (theVisibleSum > 6) then
         die2:=diceRoll2():theSum:=theSum+die2:
      end: //moves is a list of possible combos of tile flips
      moves:=combinat::partitions::list(theSum,MaxPart=9,MaxSlope=-1,MaxLength=2):
      number_of_possible_moves:=nops(moves):
      tiles_flipped:=FALSE:
      for i from 1 to number_of_possible_moves do //the code cycles through all
      //possible moves (based on dice sum) until one that can be made is found
          flips:=op(moves,i):   // flips=ith possible move (a list)
          flips_available:=1:   // Set flips_available = TRUE (i.e. 1)
          number_of_flips:=nops(flips):    // The number of integers in the list
          for j from 1 to number_of_flips do
             theFlip:=op(flips,j):   // examine integers individually
             flips_available:=flips_available*(theTiles[theFlip,2]):
          end;
          if (flips_available=1) then     //if 1, all flips were "available"
             for j from 1 to number_of_flips do
                theFlip:=op(flips,j):theTiles[theFlip,2]:=0:
             end:    //make newly flipped numbers"unavailable" (i.e. 0)
             tiles_flipped:=TRUE:break:
          end:
          if ((tiles_flipped=FALSE) and (i=number_of_possible_moves)) then
             theVisibleSum:=0;
             for k from 1 to 9 do  // Calculate the sum of all unflipped tiles.
                theVisibleSum:=theVisibleSum+theTiles[k,1]*theTiles[k,2]:
             end;
             move_is_possible:=FALSE:
             theScores[theVisibleSum+1]:=theScores[theVisibleSum+1]+1:
             //Store final score in theScores array.
          end: // end if statement
      end: // end for statement
  end: // end while
end: // end for
plot(plot::Bars2d(theScores),Header = "Frequency of Scores"):
print(Unquoted,"The probability of shutting the box was
".expr2text(theScores[1]/numtrials)):DIGITS:=7:
print(Unquoted,"Approximately = ".expr2text(float(theScores[1]/numtrials))):
```

Appendix C

Hand-Calculated Probabilities for Configurations 511, 510, 509, and 508

Configuration 511:

Since the box is already shut, the probability of shutting the box from this board is 100%. A probability of 1 is stored for this configuration.

Configuration 510:

Possible dice sum	Probability of rolling dice sum	Results of optimal "flipping"	Probability of "shutting box" after "flipping"	Probability of "shutting box" with dice sum
1	1/6		1	(1/6)*1=1/6
2	1/6	No flips possible. Game lost.	0	(1/6)*0=0
3	1/6	No flips possible. Game lost.	0	(1/6)*0=0
4	1/6	No flips possible. Game lost.	0	(1/6)*0=0
5	1/6	No flips possible. Game lost.	0	(1/6)*0=0
6	1/6	No flips possible. Game lost.	0	(1/6)*0=0
			Total Probability:	1/6+0+0+0+0+0=1/6

Configuration 509: 

Possible dice sum	Probability of rolling dice sum	Results of optimal "flipping"	Probability of "shutting box" after "flipping"	Probability of "shutting box" with dice sum
1	1/6	No flips possible. Game lost.	0	(1/6)*0=0
2	1/6		1	(1/6)*1=1/6
3	1/6	No flips possible. Game lost.	0	(1/6)*0=0
4	1/6	No flips possible. Game lost.	0	(1/6)*0=0
5	1/6	No flips possible. Game lost.	0	(1/6)*0=0
6	1/6	No flips possible. Game lost.	0	(1/6)*0=0
			Total Probability:	0+1/6+0+0+0+0=1/6

Configuration 508: 

Possible dice sum	Probability of rolling dice sum	Results of optimal "flipping"	Probability of "shutting box" after "flipping"	Probability of "shutting box" with dice sum
1	1/6		1/6	(1/6)*(1/6)=1/36
2	1/6		1/6	(1/6)*(1/6)=1/36
3	1/6		1	(1/6)*1=1/6
4	1/6	No flips possible. Game lost.	0	(1/6)*0=0
5	1/6	No flips possible. Game lost.	0	(1/6)*0=0
6	1/6	No flips possible. Game lost.	0	(1/6)*0=0
			Total Probability:	1/36+1/36+1/6+0+0+0=2/9

Appendix D

Output generated from `shutbox_dynamic.cpp` code

```
                                       BEST MOVE TO CONTINUE GAME WHEN n IS ROLLED
Config     P(Shut)    n=1        n=2        n=3        n=4        n=5        n=6        n=7        n=8        n=9        n=10       n=11       n=12
#########  1.000000   ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------
1########  0.166667   #########  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------
#2#######  0.166667   ---------  #########  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------
##3######  0.166667   ---------  ---------  #########  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------
12#######  0.222222   #2#######  1########  #########  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------
###4#####  0.166667   ---------  ---------  ---------  #########  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------
1#3######  0.222222   ##3######  ---------  1########  #########  ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------
####5####  0.166667   ---------  ---------  ---------  ---------  #########  ---------  ---------  ---------  ---------  ---------  ---------  ---------
1##4#####  0.222222   ###4#####  ---------  ---------  1########  #########  ---------  ---------  ---------  ---------  ---------  ---------  ---------
#23######  0.222222   ---------  ##3######  #2#######  ---------  #########  ---------  ---------  ---------  ---------  ---------  ---------  ---------
#####6###  0.166667   ---------  ---------  ---------  ---------  ---------  #########  ---------  ---------  ---------  ---------  ---------  ---------
1###5####  0.222222   ####5####  ---------  ---------  ---------  1########  #########  ---------  ---------  ---------  ---------  ---------  ---------
#2#4#####  0.222222   ---------  ###4#####  ---------  #2#######  ---------  #########  ---------  ---------  ---------  ---------  ---------  ---------
123######  0.333333   #23######  1#3######  12#######  #2#######  1########  #########  ---------  ---------  ---------  ---------  ---------  ---------
   ...        ...        ...        ...        ...        ...        ...        ...        ...        ...        ...        ...        ...        ...
   ...        ...        ...        ...        ...        ...        ...        ...        ...        ...        ...        ...        ...        ...

##34567#9  0.036767   ---------  ---------  ###4567#9  ##3#567#9  ##34#67#9  ##345#7#9  ##3456##9  ###4#67#9  ##34567##  ##3#5#7#9  ##3#56##9  ###4567##
12#4567#9  0.083372   ---------  1##4567#9  ###4567#9  12##567#9  12#4#67#9  12#45#7#9  12#456##9  1##45#7#9  12#4567##  #2#4567##  1##4567##  12#4#6##9
1#345678#  0.094705   ---------  ---------  1##45678#  1#3#5678#  1#34#678#  1#345#78#  1#3456#8#  1#34567##  ##34567##  1#3#5#78#  1##4567##  1#3#567##
####56789  0.010360   ---------  ---------  ---------  ---------  #####6789  ####5#789  ####56#89  ####567#9  ####5678#  ---------  ######789  #####6#89
1##4#6789  0.032440   ---------  ---------  ---------  1####6789  #####6789  1##4##789  1##4#6#89  1##4#67#9  1##4#678#  ###4#678#  1####6#89  1####67#9
#23##6789  0.031887   ---------  ##3##6789  #2###6789  ---------  #####6789  #23###789  #23##6#89  #23##67#9  #23##678#  #2###6#89  ##3##678#  #2###678#
#2#45#789  0.028481   ---------  ###45#789  ---------  #2##5#789  #2#4##789  ####5#789  #2#45##89  #2#45#7#9  #2#45#78#  ###45#7#9  ###45#78#  #2#4###89
123#5#789  0.070469   ---------  1#3#5#789  12##5#789  #2##5#789  123###789  #23###789  123#5##89  123#5#7#9  123#5#78#  1#3#5#7#9  1#3#5#78#  12##5#78#
##3456#89  0.027079   ---------  ---------  ###456#89  ##3#56#89  ##34#6#89  ##345##89  ####56#89  ##3456##9  ##3456#8#  ##3#5##89  ##34###89  ###456#8#
12#456#89  0.068764   ---------  1##456#89  ###456#89  12##56#89  12#4#6#89  12#45##89  1##4#6#89  12#456##9  12#456#8#  #2#456#8#  1##456#8#  12##56##9
1#34567#9  0.074048   ---------  ---------  1##4567#9  1#3#567#9  1#34#67#9  1#345#7#9  1#3456##9  1##4#67#9  1#34567##  ##34567##  1#3#56##9  1##4567##
#2345678#  0.086943   ---------  ##345678#  #2#45678#  #23#5678#  #234#678#  #2345#78#  #23456#8#  #234567##  #23##678#  ##34567##  #2#4567##  #23#567##
1###56789  0.025599   ---------  ---------  ---------  ---------  1####6789  1###5#789  1###56#89  1###567#9  1###5678#  ####5678#  1#####789  1####6#89
#2#4#6789  0.025903   ---------  ###4#6789  ---------  #2###6789  ---------  #2#4##789  #2#4#6#89  #2#4#67#9  #2#4#678#  #2####789  ###4#678#  #2###67#9
123##6789  0.063774   ---------  1#3##6789  12###6789  #2###6789  1####6789  123###789  123##6#89  123##67#9  123##678#  #23##678#  1#3##678#  12###678#
##345#789  0.021381   ---------  ---------  ###45#789  ##3#5#789  ##34##789  ---------  ##345##89  ##345#7#9  ##345#78#  ###45##89  ###45#7#9  ###45#78#
12#45#789  0.063437   ---------  1##45#789  ###45#789  12##5#789  12#4##789  #2#4##789  12#45##89  12#45#7#9  12#45#78#  #2#45#78#  1##45#78#  12##5#7#9
1#3456#89  0.057793   ---------  ---------  1##456#89  1#3#56#89  1#34#6#89  1#345##89  1###56#89  1#3456##9  1#3456#8#  ##3456#8#  1#34###89  1##456#8#
#234567#9  0.065065   ---------  ##34567#9  #2#4567#9  #23#567#9  #234#67#9  #2345#7#9  #23456##9  #2#4#67#9  #234567##  #23#5#7#9  ##34567##  #2#4567##
12345678#  0.145017   ---------  1#345678#  12#45678#  123#5678#  1234#678#  12345#78#  123456#8#  1234567##  #234567##  1#34567##  12#4567##  123#567##
#2##56789  0.022334   ---------  ####56789  ---------  ---------  #2###6789  #2##5#789  #2##56#89  #2##567#9  #2##5678#  ####567#9  #2####789  #2###6#89
##34#6789  0.022077   ---------  ---------  ###4#6789  ##3##6789  ---------  ##34##789  ##34#6#89  ##34#67#9  ##34#678#  ##3###789  ###4#67#9  ###4#678#
12#4#6789  0.057659   ---------  1##4#6789  ###4#6789  12###6789  #2###6789  12#4##789  12#4#6#89  12#4#67#9  12#4#678#  #2#4#678#  1##4#678#  12###67#9
1#345#789  0.051369   ---------  ---------  1##45#789  1#3#5#789  1#34##789  ##34##789  1#345##89  1#345#7#9  1#345#78#  ##345#78#  1##45#7#9  1##45#78#
#23456#89  0.053167   ---------  ##3456#89  #2#456#89  #23#56#89  #234#6#89  #2345##89  #2##56#89  #23456##9  #23456#8#  ##3456##9  ##3456#8#  #2#456#8#
1234567#9  0.118771   ---------  1#34567#9  12#4567#9  123#567#9  1234#67#9  12345#7#9  123456##9  12#4#67#9  1234567##  #234567##  1#34567##  12#4567##
##3#56789  0.018713   ---------  ---------  ####56789  ---------  ##3##6789  ##3#5#789  ##3#56#89  ##3#567#9  ##3#5678#  ####56#89  ##3###789  ####5678#
12##56789  0.046945   ---------  1###56789  ####56789  ---------  12###6789  12##5#789  12##56#89  12##567#9  12##5678#  #2##5678#  1###5678#  12###6#89
1#34#6789  0.048022   ---------  ---------  1##4#6789  1#3##6789  ##3##6789  1#34##789  1#34#6#89  1#34#67#9  1#34#678#  ##34#678#  1##4#67#9  1##4#678#
#2345#789  0.046173   ---------  ##345#789  #2#45#789  #23#5#789  #234##789  ##3#5#789  #2345##89  #2345#7#9  #2345#78#  #2#45##89  ##345#78#  #2#45#78#
123456#89  0.098168   ---------  1#3456#89  12#456#89  123#56#89  1234#6#89  12345##89  1#34#6#89  123456##9  123456#8#  #23456#8#  1#3456#8#  12#456#8#
###456789  0.015101   ---------  ---------  ---------  ####56789  ###4#6789  ###45#789  ###456#89  ###4567#9  ###45678#  ####5#789  ###4##789  ####567#9
1#3#56789  0.043140   ---------  ---------  1###56789  ####56789  1#3##6789  1#3#5#789  1#3#56#89  1#3#567#9  1#3#5678#  ##3#5678#  1###567#9  1#3##6#89
#234#6789  0.043235   ---------  ##34#6789  #2#4#6789  #23##6789  ###4#6789  #234##789  #234#6#89  #234#67#9  #234#678#  #23###789  ##34#678#  #2#4#678#
12345#789  0.089972   ---------  1#345#789  12#45#789  123#5#789  1234##789  1#3#5#789  12345##89  12345#7#9  12345#78#  #2345#78#  1#345#78#  12#45#78#
1##456789  0.035270   ---------  ---------  ---------  1###56789  1##4#6789  1##45#789  1##456#89  1##4567#9  1##45678#  ###45678#  1##4##789  1###567#9
#23#56789  0.035461   ---------  ##3#56789  #2##56789  ---------  #23##6789  #23#5#789  #23#56#89  #23#567#9  #23#5678#  ##3#567#9  ##3#5678#  #2##5678#
1234#6789  0.083159   ---------  1#34#6789  12#4#6789  123##6789  1##4#6789  1234##789  1234#6#89  1234#67#9  1234#678#  #234#678#  1#34#678#  12#4#678#
#2#456789  0.031084   ---------  ###456789  ---------  #2##56789  #2#4#6789  #2#45#789  #2#456#89  #2#4567#9  #2#45678#  #2##5#789  ###45678#  #2##567#9
123#56789  0.073866   ---------  1#3#56789  12##56789  #2##56789  123##6789  123#5#789  123#56#89  123#567#9  123#5678#  #23#5678#  1#3#5678#  12##5678#
##3456789  0.026640   ---------  ---------  ###456789  ##3#56789  ##34#6789  ##345#789  ##3456#89  ##34567#9  ##345678#  ##3#5#789  ###4567#9  ###45678#
12#456789  0.066163   ---------  1##456789  ###456789  12##56789  12#4#6789  12#45#789  12#456#89  12#4567#9  12#45678#  #2#45678#  1##45678#  12##567#9
1#3456789  0.057121   ---------  ---------  1##456789  1#3#56789  1#34#6789  1#345#789  1#3456#89  1#34567#9  1#345678#  ##345678#  1##4567#9  1##45678#
#23456789  0.051732   ---------  ##3456789  #2#456789  #23#56789  #234#6789  #2345#789  #23456#89  #234567#9  #2345678#  ##34567#9  ##345678#  #2#45678#
123456789  0.091253   ---------  #23456789  1#3456789  12#456789  123#56789  1234#6789  12345#789  123456#89  1234567#9  12345678#  #2345678#  1#345678#
```