James V. Rauff

Activities

A Millennium Prize Problem for Students

This million-dollar prize in mathematics gives mathematics teachers a unique opportunity to devise interesting activities

On May 24, 2000, the Clay Mathematics Institute (www.claymath.org) announced its seven Millennium Prize Problems. The Institute will award $1,000,000 for the solution to any of these problems. Each problem is significant in its own area of mathematics and has resisted solution for some time.

In an era when the average NBA player's salary is $4.5 million and when someone with a broad knowledge of trivia can win a million dollars on a televised game show, this million-dollar prize in mathematics gives mathematics teachers a unique opportunity to devise and use interesting activities. Unfortunately, most of the Millennium Prize Problems are difficult for nonmathematicians to understand and are completely incomprehensible to children. However, one problem is accessible to middle and high school students. This article provides guided lessons that will help students understand, although probably not solve, the Millennium Prize Problem known as *P versus NP*.

P VERSUS NP

The P versus NP problem arises in computer science in the study of algorithms for decision problems, which are problems that demand a yes or no answer. The following are some sample decision problems:

- Can we drive from Chicago to Louisville using only interstate highways?
- Is this C++ program free of syntax errors?
- Will we have colonies on Mars in twenty-five years?
- Given a finite set of words and an $n \times n$ matrix of black and white squares, can a crossword puzzle be made using all the words?

Certain decision problems can be solved with a computer algorithm. For example, when a C++ compiler rejects code, it has made a decision that the code is not grammatical C++. A computer equipped with a representation of the interstate highway system could answer a question about getting to Louisville from Chicago on interstate highways. However, no computer algorithm can answer a question about future colonies on Mars.

Computer algorithms for decision problems are frequently assessed and compared with respect to the amount of time they take to find an answer. The time is usually expressed as a function of the size of the input. For example, suppose that the decision problem is "Here is a list of numbers. Is the largest number in this list greater than 2001?" One possible computer algorithm is the following:

Start with the first number in the list.

1. Compare the number to 2001.
2. If the number is less than or equal to 2001, go to step 3; otherwise, answer yes and stop.
3. If the list includes another number to examine, go to step 1; otherwise, answer no and stop.

If none of the numbers in the list is greater than 2001, the algorithm looks at every number in the list. If a time t seconds is needed to compare a number to 2001 and if n numbers are in the list, the algorithm will run for at most nt seconds before it

Jim Rauff, jrauff@millikin.edu, teaches mathematics and computer science at Millikin University, Decatur, IL 62522. He is interested in formal language theory and the history of mathematics.

Edited by ***Eileen Schoaff,*** *eileen@schoaff.org, professor emeritus, Buffalo State College, Buffalo, NY 14222*

This section is designed to provide in reproducible formats mathematics activities appropriate for students in grades 7–12. This material may be reproduced by classroom teachers for use in their own classes. Readers who have developed successful classroom activities are encouraged to submit manuscripts, in a format similar to the "Activities" already published, to the senior journal editor for review. Of particular interest are activities focusing on the Council's curriculum standards, its expanded concept of basic skills, problem solving and applications, and the uses of calculators and computers.

Write to NCTM, attention: infocentral, or send e-mail to infocentral@nctm.org, for the catalog of educational materials, which lists compilations of "Activities" in bound form.—Ed.

stops. The algorithm's time complexity is a linear function of the size of its input. The decision problem is then solvable by the algorithm in linear time.

If a decision problem is solvable by some algorithm in polynomial time and solvable by another algorithm in exponential time, then the polynomial-time algorithm is considered superior to the exponential-time algorithm because of what is called the *combinatorial explosion.* **Table 1** shows quite dramatically this explosion for an algorithm whose time is n^2 nanoseconds (polynomial time) versus one whose time is 2^n nanoseconds (exponential time).

The student explorations in this article deal primarily with this combinatorial explosion. The goal is to give students an appreciation for the significant differences between polynomial and nonpolynomial time. If a decision problem can be *solved* by a polynomial-time algorithm, then that decision problem is a member of the class P.

Some very famous decision problems have yet to be solved with a polynomial-time algorithm. We do not know whether they can be solved with a polynomial-time algorithm that nobody has yet found or whether finding one is impossible. The most famous of these problems is the Hamiltonian circuit problem (HCP):

> Given a road map, does a driving tour exist that starts and ends at the same city and visits every other city on the map exactly one time?

The activity sheets examine this problem. The time complexity is determined by counting the number of possible orderings of the cities visited. No polynomial-time algorithm for solving the HCP is known. However, if someone gave us a route through the map, we could check to see whether every city was visited exactly once—and we could check it rather quickly.

The class of decision problems for which potential answers can be *checked* by some algorithm in a number of steps that is a *polynomial* function of the size of the input is called the *class NP.* Some NP problems have the property that all other NP problems can be converted into them. These conversions are all achieved by algorithms in a number of steps that is a polynomial function of the size of the input. These problems are called *NP-complete* and in a way, they represent all of NP. Hundreds of these NP-complete problems exist (see Garey and Johnson [1979]). If any NP-complete problem can be shown to be in P, then all NP problems are in P. To solve the NP problem, we would have to convert it to this NP-complete problem and then solve the NP-complete problem. The conversion algorithm and the solution are achieved in a number of steps that is a polynomial function of the input.

If anyone proves that an NP-complete problem is, in fact, in P, the implications for computer science would be tremendous. Thousands of extremely difficult decision problems would gain polynomial-time algorithms. Every corner of the computer science community would be affected.

TABLE 1
Illustration of the Combinatorial Explosion

Input size	$n = 10$	$n = 100$
time = n^2	0.0000001 seconds	0.00001 seconds
time = 2^n	0.000001024 seconds	4.019×10^{13} years

However, most computer scientists and mathematicians do not believe that any of the NP-complete problems are actually P problems, but nobody has been able to prove this belief one way or another. This proof is the P versus NP problem that has a \$1,000,000 bounty on its head.

THE ACTIVITIES

The activities take the student on a tour of computational complexity that leads to the statement of the P versus NP problem. I have used the activities with individual students and with teams of three. The material is accessible to high school students, as well as middle school students who have taken some algebra.

All the lessons discuss people who inhabit a country ruled by a fictional queen who commissions annual birthday gifts for her daughter. The time required to make the gifts is a function of the age of the daughter. Each gift is designed to express a different function (quadratic, cubic, and exponential). In **activity 1,** the student investigates the time complexity of the three gifts and discovers how exponential functions can greatly surpass polynomial functions. **Activity 2** looks at space complexity for the same gifts. **Activity 2** is not needed before doing **activity 3.** My students wanted to look at the space needed to make the gifts, so I added some investigations that I based on their interests. **Activity 3** adds the birthday tour (a variation of HCP) to the lessons and describes the P versus NP problem.

If anyone proves that an NP-complete problem is in P, the implications for computer science would be tremendous

SOLUTIONS

Sheet 1

1. See **table 2.**

2. $m = n^2$.

TABLE 2
Sheet 1, Question 1

Theta's Age (Years)	Number of Tiles in the Square Mosaic
1	1
2	4
3	9
4	16
5	25
6	36
10	100
15	225
16	256

3. See **table 3.**
4. $c = n^3$.
5. See **table 4.**
6. $b = 2^{n-1}$.
7. See **table 5.**
8. They all increase, but the bead stringer's time increases much faster.
9. Mosaic: $t = 3 \bullet n^2$

   Cake: $t = 0.5 \bullet n^3$

   Necklace: $t = \frac{s}{60} \bullet 2^{n-1} = \frac{1}{12} \bullet 2^{n-1}$
10. Because stringing time eventually becomes longer than tiling or baking time, the royal baker probably would not want to switch with the royal bead stringer.

The activities take the student on a tour of computational complexity

TABLE 3
Sheet 1, Question 3

Theta's Age (Years)	Number of Cubecakes in the Birthday Cake
1	1
2	8
3	27
4	64
5	125
6	216
10	1,000
15	3,375
16	4,096

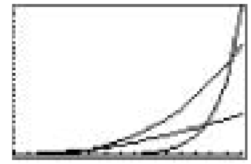
TABLE 4
Sheet 1, Question 5

Theta's Age (Years)	Number of Beads in the Necklace
1	1
2	2
3	4
4	8
5	16
6	32
10	512
15	16,384
16	32,768

TABLE 5
Sheet 1, Question 7

Theta's Age (Years)	Time to Make Mosaic	Time to Assemble Cake	Time to Make Necklace
1	3 min.	30 sec.	5 sec.
2	12 min.	4 min.	10 sec.
3	27 min.	13.5 min.	20 sec.
4	48 min.	32 min.	40 sec.
5	75 min. = 1 hr., 15 min.	1 hr., 2.5 min.	1 min., 20 sec.
6	108 min. = 1 hr., 48 min.	1 hr., 48 min.	2 min., 40 sec.
10	300 min. = 5 hrs.	8 hrs., 20 min.	42 min., 40 sec.
15	11 hrs., 15 min.	28 hrs., 7.5 min.	22 hrs., 45 min., 20 sec.
16	12 hrs., 48 min.	34 hrs., 8 min.	45 hrs., 30 min., 40 sec.

Sheet 2

1. 

   The window is x-axis [1, 16], y-axis [0, 2750]. Although the graph shows a continuous function, the graphs are discrete functions, since the gifts were done only for the birthdays.
2. When Theta was ten years old, the bead stringer worked the shortest time. When she was sixteen, the tiler worked the shortest time.
3. For $n = 1$ to 5, the tiler's time was higher than the others, the bead stringer's time was lower than the others, and the baker's time was higher than the bead stringer's time but lower than the tiler's time. For $n = 6$, the mosaic and the cake require the same amount of time, which is higher than the amount of time required by the bead stringer. For $n = 7$ to 13, the baker's time was higher than the others, the bead stringer's time was lower than the other two, and the tiler's time was higher than the bead stringer's time but lower than the baker's time. For $n = 14$ to 15, the baker's time was higher than the others, the tiler's time was lower than the other two, and the bead stringer's time was higher than the tiler's time but lower than the baker's time. For $n > 15$, the bead stringer's time was higher than the others, the tiler's time was lower than the other two, and the baker's time was higher than the tiler's time but lower than the bead stringer's time.
4. See **table 6.**

TABLE 6
Sheet 2, Question 4

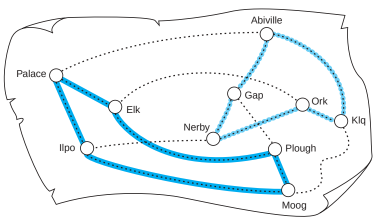
Theta's Age	Area of Mosaic	Volume of Birthday Cake	Length of Necklace (Beads Only)
1	1 sq. in.	8 cu. in.	1/4 in.
2	4 sq. in.	64 cu. in.	1/2 in.
3	9 sq. in.	216 cu. in.	1 in.
4	16 sq. in.	512 cu. in.	2 in.
5	25 sq. in.	1,000 cu. in.	4 in.
6	36 sq. in.	1,728 cu. in.	8 in.
10	100 sq. in.	8,000 cu. in.	128 in.
15	225 sq. in.	27,000 cu. in.	4,096 in.
16	256 sq. in.	32,768 cu. in.	8,192 in.
n	n^2 sq. in.	$8n^3$ cu. in.	$\frac{1}{4} \bullet 2^{n-1}$ in.

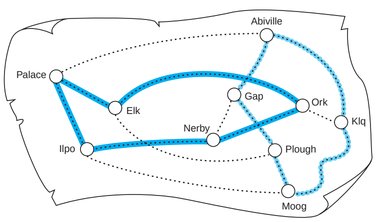
5. On her twentieth birthday, it will be 131,072 inches long.
6. It is 10,922 2/3 feet, or 2 34/495 miles.
7. The bead stringer could combine all the necklaces made in previous years.

Sheet 3

1. Palace-Cyr-Abiville-Belloz-Didgery-Palace
   Palace-Cyr-Didgery-Belloz-Abiville-Palace
   Palace-Didgery-Belloz-Abiville-Cyr-Palace
   Palace-Abiville-Belloz-Didgery-Cyr-Palace
2. If Pa represents the palace, the first letter of each town represents the town, and a road is denoted by the towns it connects, three possible tours are the following: PaACEDBGJKFHLIPa, PaILHFKJGBDECAPa, PaECABDGJKFHLIPa
3. None. We must use exactly two of the three roads entering each town—one road to enter the town and one road to leave the town.

   The tour must include two of the roads PaA, PaE, and PaI. If PaA is not included, then AK and AG must be. Only one of GP and GN is therefore included. If GN is used, then we must have EP and PM, since two roads must go through each town. We then must use ON and OK. Thus, IN is excluded, so we must use IM and exclude MK. The result is two unconnected loops: PaEPMIPa and AKONGA. So we go back and exclude GN and include GP. We must then have NO and NI. IM is then excluded, and MP and MK must be included. Thus, OK is excluded and EO is required. Again we have separate loops: PaINOEPa and AGPMKA. Excluding PaE or PaI initially leads to similar results.





4. PaZWAPa, PaZAWPa, PaAWZPa, PaAZWPa PaWZAPa, and PaWAZPa
5. 5! = 120
6. See **table 7.**

TABLE 7
Sheet 3, Question 7

Number of Cities	Number of Potential Tours
3	3! = 6
5	5! = 120
10	10! = 3,628,800
12	12! = 479,001,600
20	$20! = 2.4329 \times 10^{18}$
n	$n!$

TABLE 8
Sheet 3, Question 8

Number of Cities	Number of Potential Tours	Time to Calculate Each Tour	Time to Calculate All Potential Tours
5	5! = 120	0.000000005 sec.	0.0000006 sec.
10	10! = 3,628,800	0.00000001 sec.	0.036288 sec.
15	15!	0.000000015 sec.	19,615.11552 sec. = 5 hrs., 26 min., 55 sec.
20	$20! = 2.4329 \times 10^{18}$	0.00000002 sec.	1,543 years
30	30!	0.00000003 sec.	2.523×10^{17} years
n	$n!$	$n(10^{-9})$ sec.	$n(10^{-9})n!$ sec.

8. See **table 8.**

BIBLIOGRAPHY

Cook, Stephen. "The P versus NP Problem." Clay Mathematics Institute. www.claymath.org /prizeproblems/p_vs_np.pdf. World Wide Web.

Garey, Michael R., and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York: W. H. Freeman, 1979.

Jackson, Allen. "Million-Dollar Mathematics Prizes Announced." *Notices of the American Mathematical Society* 47 (September 2000): 877–79.

Kaye, Richard. "Minesweeper Is NP-Complete." *The Mathematical Intelligencer* 22 (spring 2000): 9–15.

———. "Some Minesweeper Configurations." Manuscript dated August 13, 2000. Available at www.mat.bham.ac.uk/R.W.Kaye/minesw /minesw.pdf. World Wide Web.

Kinber, Efim, and Carl Smith. *Theory of Computing: A Gentle Introduction.* Upper Saddle River, N.J.: Prentice-Hall, 2001.

(Worksheets begin on page 30)

The queen of the wonderful land of Complexis was ecstatic about the birth of her daughter, Theta. She decided to plan annual birthday celebrations for her. The queen called in her three royal artists and gave them each an assignment to be fulfilled every year in time for Theta's birthday.

The queen told the royal tiler to make a square mosaic of square iridescent tiles each year. The number of tiles on each side of the mosaic must equal the age of Theta on her birthday.

1. Complete the table for the number of tiles in the mosaic.

Theta's Age (Years)	Number of Tiles in the Square Mosaic
1	1
2	4
3	9
4	
5	
6	
10	
15	
16	

2. Write a formula for the number of tiles in the mosaic on Theta's nth birthday. Let m represent the number of tiles in the mosaic on Theta's nth birthday.

The queen told the royal baker to make a large cake in the shape of a cube. She wanted him to build the cake from small cube-shaped cupcakes that he calls *cubecakes.* The number of cubecakes along the side of the birthday cake must equal the age of Theta on her birthday.

3. Complete the table for number of cubecakes in the birthday cake.

Theta's Age (Years)	Number of Cubecakes in the Birthday Cake
1	1
2	8
3	27
4	
5	
6	
10	
15	
16	

4. Write a formula for the number of cubecakes in the birthday cake on Theta's nth birthday. Let c represent the number of cubecakes used to make the birthday cake on Theta's nth birthday.

Finally, the queen instructed the royal bead stringer to make a necklace of gold beads for Theta. On Theta's first birthday, the necklace would have one gold bead. On her second birthday, the number of beads would double to two beads; and it would continue to double for each birthday thereafter.

5. Complete the table for number of gold beads in the necklace.

Theta's Age (Years)	Number of Beads in the Necklace
1	1
2	2
3	4
4	
5	
6	
10	
15	
16	

6. Write a formula for the number of gold beads in the necklace on Theta's nth birthday. Let b represent the number of gold beads used to make the necklace for Theta's nth birthday. (Remember that $2^0 = 1$.)

Each of the royal artists estimated the amount of time that he or she would need to assemble the gifts for Theta. Assuming that they had all the materials available ahead of time, the times were as follows:

- The royal tiler could lay one tile in the mosaic in three minutes.
- The royal baker could put one cubecake into the birthday cake in thirty seconds.
- The royal bead stringer could string one gold bead on the necklace in five seconds.

7. Complete the table for the amount of time that each royal artist would need to construct the gift for Theta.

Theta's Age (Years)	Time to Make Mosaic	Time to Assemble Cake	Time to Make Necklace
1	3 min.	30 sec.	5 sec.
2	12 min.	4 min.	10 sec.
3	27 min.	13.5 min.	20 sec.
4	48 min.	32 min.	40 sec.
5			
6			
10			
15			
16			

8. What do you notice about the time that each artist needs to complete his or her task?

9. Write a formula for the assembly time t, in minutes, of each gift on Theta's nth birthday.

Mosaic: t = ______________________________

Cake: t = ______________________________

Necklace: t = ______________________________

After Theta's seventh birthday, the royal baker and the royal tiler complained to the queen that their work required more than two hours, whereas the royal bead stringer needed only five minutes. They did not think that it was fair. The royal bead stringer, however, just chuckled about it.

The queen, being sympathetic, said that either the royal tiler or the royal baker could switch jobs with the royal bead stringer, since they were all talented and could do one another's work very well. However, if they switched, the change would be permanent.

10. If you were the royal baker, would you switch with the royal bead stringer? Explain. Before you answer, take a look at the times needed to make the birthday gifts when Theta is sixteen years old.

In the last activity, you saw that for the first few years, the amount of time needed to string the gold beads was much less than that required for making the mosaic or the cake. However, the bead-stringing time eventually became considerably longer than the tiling or baking time. The following table shows some of the times again, along with the formulas for computing them:

Theta's Age (n)	Tiling Time = $3n^2$ (Minutes)	Cake Time = $0.5n^3$ (Minutes)	Beading Time = $\left(\frac{5}{60}\right) \cdot 2^{n-1}$ (Minutes)
1	3 min.	30 sec.	5 sec.
2	12 min.	4 min.	10 sec.
5	1 hr., 15 min.	1 hr., 2 min., 30 sec.	1 min., 20 sec.
10	5 hr.	8 hr., 20 min.	42 min., 40 sec.
13	8 hr., 27 min.	18 hr., 18 min., 30 sec.	5 hr., 41 min., 20 sec.
15	11 hr., 15 min.	28 hr., 7 min., 30 sec.	22 hr., 45 min., 20 sec.
16	12 hr., 48 min.	34 hr., 8 min.	45 hr., 30 min., 40 sec.

1. Using a graphing calculator or graphing software, draw the graphs of $t = 3n^2$, $t = 0.5n^3$, and $t = (5/60) \cdot 2^{n-1}$, for n a positive integer, on the same coordinate system.

2. Who worked the shortest amount of time making a gift when Theta was ten years old?

   When she was sixteen years old?

3. Using the phrases "higher than" and "lower than," describe the behavior of the graphs. Be sure to zoom in and out to obtain a good look at the graphs.

In constructing Theta's birthday gifts, the queen's artists were each carrying out the steps in an algorithm, a step-by-step procedure to accomplish a specific task. Computer scientists are interested in the amount of time that is required for an algorithm to be executed. As you have discovered, the time required for an algorithm to be executed depends on the number of steps that make up the algorithm. The amount of time required for an algorithm to be executed, viewed as a function of the size of the task, is called the *time complexity* of the algorithm.

The times that the baker and tiler needed to complete their tasks, $t = 3n^2$ and $t = 0.5n^3$, respectively, were polynomial functions of the number of pieces they needed. Their algorithms can be performed in *polynomial time*. The bead stringer's time,

$$t = \left(\frac{5}{60}\right) \cdot 2^{n-1},$$

is an exponential function of the number of beads needed. The bead stringer can perform his algorithm in *exponential time*.

Obviously, exponential-time algorithms can become really time-consuming.

You have received a pretty good view of the time complexity of the birthday-gift algorithms. Another way of seeing the complexity of algorithms is by looking at the space that they need. In a computer, space is measured in terms of memory. This activity examines the space needed to create the birthday gifts for Theta.

The queen also determined the sizes of the birthday gifts. She commanded that each tile for the birthday mosaic be one inch square. Each cubecake must have two-inch sides. Each gold bead must be a sphere one-quarter inch in diameter.

4. Complete the table below.

Theta's Age	Area of Mosaic	Volume of Birthday Cake	Length of Necklace (Beads Only)
1	1 sq. in.	8 cu. in.	1/4 in.
2	4 sq. in.	64 cu. in.	1/2 in.
3	9 sq. in.	216 cu. in.	1 in.
4			
5			
6			
10			
15			
16			
n			

5. The length of the necklace has become much longer by the time that Theta is six years old. How long will it be on her twentieth birthday?

6. How long is it in feet? ________________ How about in miles? ________________

The space requirements for the mosaic and cake are polynomial functions of n, whereas the space requirement for the necklace is an exponential function of n.

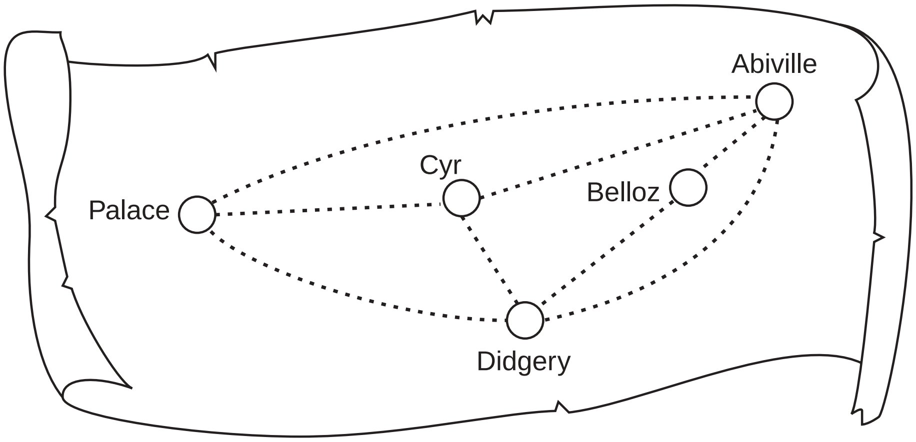
The following chart shows the time and space requirements for gifts on Theta's thirtieth birthday.

Gift	Time Needed to Make Gift	Size of Gift
Mosaic	45 hours	6.25 square feet
Cake	9 days, 9 hours	125 cubic feet
Necklace	about 85 years	about 2118 miles

The thirtieth-birthday necklace would reach from Chicago to San Diego (2105 miles), or from Boise, Idaho, to Jackson, Mississippi (2115 miles). Not only that, the royal bead stringer would need to start making the necklace fifty-five years before Theta was born and probably even before the queen was born!

7. How could the bead stringer shorten the amount of time needed to make this enormous necklace?

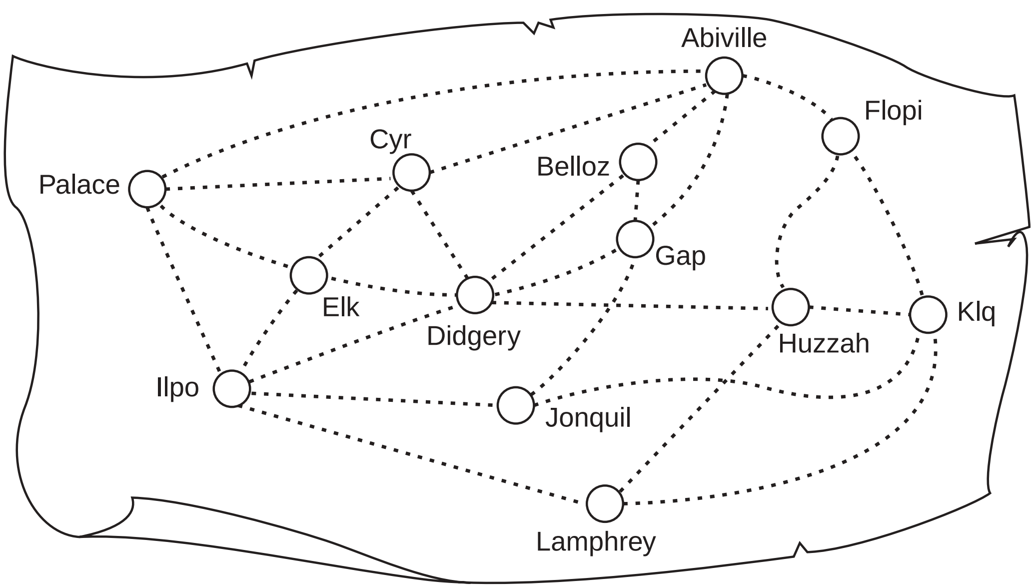
After three years, the queen decided to start another birthday tradition for Theta. The queen randomly chose four towns in her kingdom and promised to make a grand tour of them all on Theta's fourth birthday. She would start and end the tour at her palace in the capital city. The queen also commanded that each town be visited only once on any birthday tour. So each year the royal mapmaker and royal driver got together to plan the birthday tour. The map shows the tour that they prepared for Theta's fourth birthday.



The royal driver must be sure that a birthday tour is possible. That is, she must find a route that starts and ends at the palace and that visits each town exactly once.

1. Use the map to find four possible birthday tours.

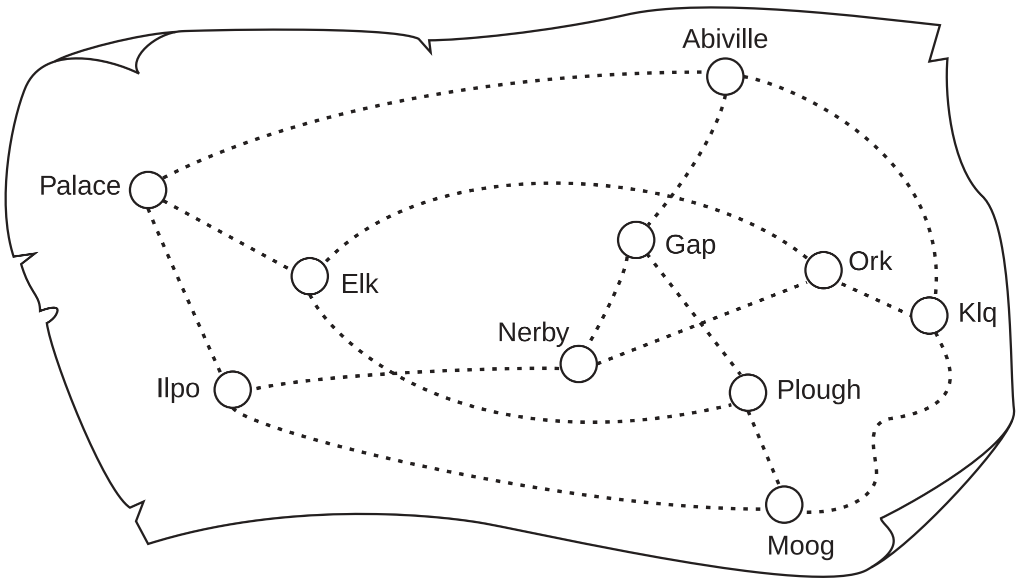
The fourth-year birthday tour was fairly easy to plan, but by the time Theta's twelfth birthday rolled around, the job became more difficult. The map shows the tour for Theta's twelfth birthday.



From the *Mathematics Teacher,* January 2003

2. Find three possible birthday tours for Theta's twelfth birthday.

3. Using the map shown, try to find a birthday tour for Theta's tenth birthday. If you cannot find a tenth-birthday tour, explain why such a tour cannot be found.



One way to be sure that no birthday tour exists for a particular n is to try every possible route.

4. Suppose that the queen had begun the touring tradition on Theta's third birthday. Then the tour only had to visit three cities. Imagine that the queen selected Abiville, Zeplch, and Whazzup. You can write a possible tour just by listing the places in the order that they would be visited. Thus, one potential tour would be palace-Zeplch-Whazzup-Abiville-palace. Another would be palace-Whazzup-Abiville-Zelpch-palace. List the six possible tours that could be made. Remember that a tour must begin and end at the palace.

How did you know that the number of potential tours was six? Each tour begins at the palace and ends at the palace, so you only need to find the number of different arrangements of the three towns in the middle. Three choices are possible for the first town in the tour; two choices are possible for the second, since the first has already been picked; and only one choice exists for the third. Thus, the number of potential tours is $3 \times 2 \times 1$, or six.

5. How many different potential tours exist for five cities? ________________________________

To avoid writing long patterned products like $5 \times 4 \times 3 \times 2 \times 1$, write instead 5!, which is pronounced as "five factorial." Mathematicians think that 16! is easier to write and read than

$$16 \times 15 \times 14 \times 13 \times 12 \times 11 \times 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1.$$

6. Complete the table.

Number of Cities	Number of Potential Tours
3	3! = 6
5	5! = 120
10	10! = 3,628,800
12	
20	
n	

Some of the potential tours are impossible if no road connects the towns; but at least when you have checked them all and have not found a tour, you know that no tour is possible.

The royal driver has a computer. She can put a map of Complexis and the towns that were chosen into the computer and ask the computer to look for tours. Her computer is fast enough to check any possible tour at a speed of one nanosecond (one billionth of a second) per town.

For the tenth-birthday map, the royal driver's computer examined all 3,628,800 potential tours and found none that fit the queen's requirements. This calculation took only $10 \times 10^{-9} = 10^{-8}$ seconds per tour, or 0.036288 seconds.

7. Complete the table.

Number of Cities	Number of Potential Tours	Time to Calculate Each Tour	Time to Calculate All Potential Tours
5	5!	0.000000005 sec.	0.0000006 sec.
10	10!	0.00000001 sec.	0.036288 sec.
15			
20			
30			
n	$n!$	$n(10^{-9})$ sec.	$n(10^{-9})n!$ sec.

On Theta's twentieth birthday, twenty cities are in the tour. The royal driver's computer takes only twenty nanoseconds to *check a tour*. Yet, the computer needs about 48,658,040,160 seconds, or approximately 1542 years to *check all the potential tours*. If no tour was possible, the royal driver would have to wait 1542 years for her computer to finish checking every possibility before she would know for sure. Theta would be well beyond twenty years old by then.

So the problem of deciding whether a birthday tour is possible is considerably more time-consuming than the process of checking whether a specific sequence of cities can be used for a birthday tour.

Some problems, like checking a birthday tour, or algorithms, like making the birthday cake, can be solved or finished in times that are polynomial functions of the size of their input. Checking the $\boldsymbol{n}$th birthday tour only took $10^{-9}\boldsymbol{n}$ seconds, and assembling the $\boldsymbol{n}$th birthday cake took $0.5\boldsymbol{n}^3$ minutes; $10^{-9}\boldsymbol{n}$ is a first-degree polynomial (linear) function of $\boldsymbol{n}$, and $0.5\boldsymbol{n}^3$ is a third-degree polynomial function of $\boldsymbol{n}$.

Other problems, like searching all the potential birthday tours or making the gold necklace, require times that are not polynomial functions of the size of their input. Making the $\boldsymbol{n}$th gold necklace took $(5/60)2^{n-1}$ minutes (exponential time), and the complete search for a birthday tour took a maximum of $10^{-9}\boldsymbol{n}(\boldsymbol{n}!)$ seconds (factorial time).

Problems that can be solved in a time that is a polynomial function of the size of their inputs are called *P problems.* These problems usually can be solved within a reasonable amount of time. Checking a potential birthday tour is a P problem.

Problems that have solutions that are not P problems are called *intractable* because the time necessary to solve them quickly becomes enormous—often measured in millennia. You do not have the time to wait around for these solutions. Searching all routes for a birthday tour is intractable because finding all solutions would take millennia for $\boldsymbol{n}$ greater than 20.

But even though *finding* a birthday tour may not be a P problem, *checking* an answer to the birthday problem is a P problem. Problems whose answers can be checked in polynomial time but have no known solution in polynomial time are called *NP problems*.

The birthday-tour problem is also a special kind of NP problem, called an *NP-complete problem*. NP-complete problems have a special relationship to all the other NP problems. If some clever mathematician or computer scientist finds a polynomial-time solution to the birthday-tour problem (or any other NP-complete problem), then finding a polynomial-time solution to every NP problem is possible and therefore all NP problems are actually P problems.

Most computer scientists and mathematicians think that such a solution will never be found. They think that NP problems are not P problems, but nobody has been able to prove this possibility. These problems show up in scheduling delivery-truck routes, backing up files on floppy disks, assigning students to classes, and making crossword puzzles.

Computer scientists are very interested in finding out once and for all whether every NP problem is in fact a P problem. This question is known as the *P versus NP problem*. The answer to the question is so important that the Clay Mathematics Institute (www.claymath.org) has offered a $1,000,000 prize to anyone who can answer the question one way or another. Who wants to be a millionaire?