2014 IUPUI HIGH SCHOOL MATH CONTEST PROBLEMS

(1) Terence's homework consists of computing the cube roots of 50 different numbers. These are all 5- or 6-digit numbers, and their cube roots are integers. Unfortunately, the notebook where he wrote the problems got wet in the rain and all the digits in the 1s, 10s, and 100s places are blurred. How can Terence still solve his homework with the information left?

(2) Ms. Puckett lives in Nemesis, an asteroid in the form of a regular dodecahedron with edges 100 miles long. Her house is in one of the 20 vertices, and she wants to network with a friend that lives in the diametrically opposite vertex. What is the shortest distance she needs to drive in order to reach her friend? (Ms. Puckett does not need to stay on the edges, but can drive anywhere through the faces of the dodecahedron)

(3) Show that for any fixed n it is possible to reach every number in $0, 1, ..., n$ by the following method: start with n and repeatedly apply one of the functions $f(x) = n - x$ and $g(x) = \lfloor x/2 \rfloor$ (take half and round down). For example, when $n = 5$ we can reach 2, 1, and 0 by applying g three times. Then, we can reach 3 and 4 by applying f to 2 and 1, respectively.

(4) Let P be a convex potato floating in 3-dimensional space (convex means that the line segment between any two points of P lies within P). We are told that no matter how P is rotated, the points directly below P always form a perfectly round shadow. With this information show that P is a perfectly round potato.

(5) You have ten Scrabble© tiles with the letters A E F G I L M N O S in that order. Your task is to shuffle them around to obtain a ten-letter English word (there is only one). However, you are only allowed two types of manipulation: you can shift the first letter to the last position, and you can pick the letters in odd positions (in order) followed by the letters in even positions (in order). Thus, starting with the letters in alphabetic order, you can reach E F G I L M N O S A, and A F I M O E G L N S. What is the shortest sequence of moves needed to complete the task?

(6) Write an essay of 500 to 700 words (complete with references) on an application of mathematics to social networks.

2014 IUPUI HIGH SCHOOL MATH CONTEST

1. Problem 1

Terence's homework consists of computing the cube roots of 50 different numbers. These are all 5- or 6-digit numbers, and their cube roots are integers. Unfortunately, the notebook where he wrote the problems got wet in the rain and all the digits in the 1s, 10s, and 100s places are blurred. How can Terence still solve his homework with the information left?

Terence can still complete his homework if he can uniquely identify an integer n by the digits in the 1000s and 10000s places of its cube. Terence would not be able to do this only if at least two of these integers, say $n > m$, had cubes n^3 and m^3 that are different only within the digits in 1s, 10s, and 100s places. In other words, only if

$$n^3 - m^3 < 1000.$$

Since $n > m$, it is true that $n \geq m + 1$ and therefore

$$1000 > n^3 - m^3 \geq (m+1)^3 - m^3 = 3m^2 + 3m + 1.$$

Notice also that his answers must be bigger than 21 since $21^3 = 9261$ is only a 4-digit number. That is, $m > 21$, and therefore

$$1000 > 3m^2 + 3m + 1 > 3(21)^2 + 3 \cdot 21 + 1 = 1323 + 64 = 1387,$$

which is nonsense. Thus, all the cubes of integers bigger than 21 can be uniquely identified by their digits in the 1000s and 10000s places and Terence can solve his homework by simply computing the cubes of successive integers bigger than 21 and comparing the digits in the 1000s and 10000s places of the result to the ones written in his notebook.

2. Problem 2

Ms. Puckett lives in Nemesis, an asteroid in the form of a regular dodecahedron with edges 100 *miles long. Her house is in one of the* 20 *vertices, and she wants to network with a friend that lives in the diametrically opposite vertex. What is the shortest distance she needs*

to drive in order to reach her friend? (Ms. Puckett does not need to stay on the edges, but can drive anywhere through the faces of the dodecahedron)

Let us write **n** for Ms Puckett's home vertex, and **s** for her friend's. Obviously, the first pentagonal face visited (call it A) contains **n** as one of its vertices, which leaves Ms Puckett the option of leaving A to enter face B or face C; see Figure 1 (because of the symmetry of the figure, all other options are equivalent to one of these). From either face, Ms Puckett will enter face D and head straight to **s**.

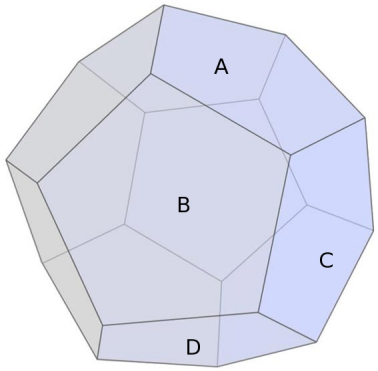


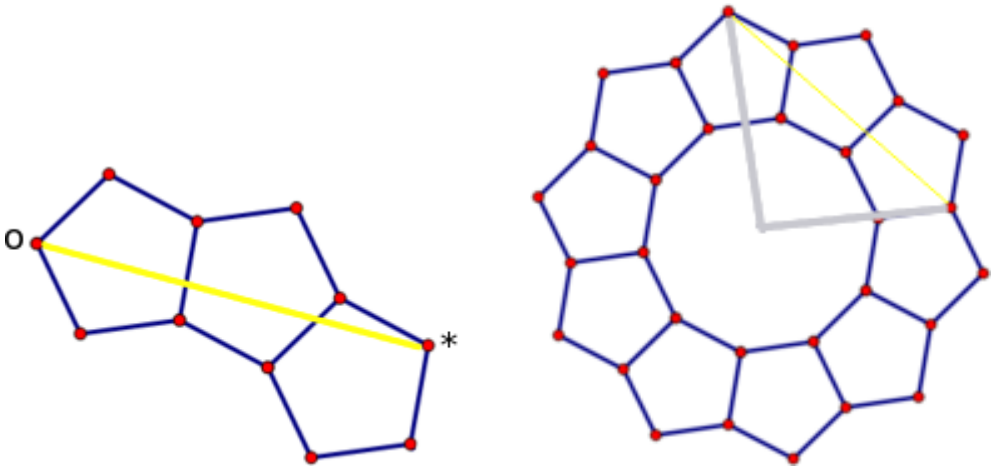
Figure 1

To find the shortest route, imagine that the surface of the dodecahedron is made of paper and that we can unfold it to form a flat map. Then the shortest path between two points on the dodecahedron will correspond to a straight line on the map. Because of the reasoning in the paragraph above, we only need to unfold three faces, either A, B, D, or A, C, D. In both cases, the map looks like Figure 2, where the vertices **n** and **s** correspond to $\circ$ and $*$ (which is which depends on whether Ms Puckett drives through B or C). The problem becomes that of computing the distance between $\circ$ and $*$.

A nice way to find this distance is to extend the chain of pentagons until it closes. We know it does because the angle between two pentagons ($360^\circ - 2 \cdot 108^\circ = 144^\circ$) is the same as the angle between two consecutive edges of the regular decagon in the center ($\frac{8}{10} \cdot 180^\circ = 144^\circ$).

The distance from the center of the figure to $*$ is the dodecagon's apothem plus the pentagon's height:

$$\frac{100}{2\tan(\frac{\pi}{10})} + \left(\frac{100}{2\tan(\frac{\pi}{5})} + \frac{100}{2\sin(\frac{\pi}{5})} \right).$$



Figures 2 and 3

The distance from the center of the figure to ∘ is the dodecagon's circumradius plus the pentagon's edge:

$$\frac{100}{2\sin(\frac{\pi}{10})}+100.$$

The final answer is given by Pythagoras:

$$\sqrt{\left(\frac{100}{2\tan(\frac{\pi}{10})}+\left(\frac{100}{2\tan(\frac{\pi}{5})}+\frac{100}{2\sin(\frac{\pi}{5})}\right)\right)^2+\left(\frac{100}{2\sin(\frac{\pi}{10})}+100\right)^2}.$$

This is about 404.0574 miles. As a curious note, other methods of computing the answer give the formula

$$100\sqrt{\frac{17+7\sqrt{5}}{2}},$$

and it takes a good deal of playing around with trigonometric identities (we will not do it here) to see that this is the exact same value as above.

3. Problem 3

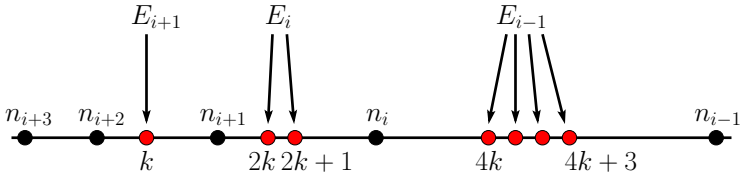
Show that for any fixed n it is possible to reach every number in $0, 1, ..., n$ by the following method: start with n and repeatedly apply one of the functions $f(x) = n-x$ and $g(x) = \lfloor x/2 \rfloor$ (take half and round down). For example, when $n = 5$ we can reach 2, 1, and 0 by applying g three times. Then, we can reach 3 and 4 by applying f to 2 and 1, respectively.

Assume to the contrary that there is some non-empty exceptional set of integers between 1 and n, say E, that cannot be reached by successive applications of functions f and g.

Let us write $n_0 = n$, $n_1 = g(n)$, $n_2 = g(n_1) = g(g(n))$, $n_3 = g(n_2) = g(g(g(n)))$, and so on. Recursively, we can write $n_{i+1} = g(n_i)$. Notice that $g(k) < k$ for any positive integer k and therefore

$$0 < 1 < \cdots < n_{i+1} < n_i < n_{i-1} < \cdots < n_1 < n_0 = n.$$

Clearly, all the integers n_i are not in the exceptional set E. Therefore, we can split E into the sets E_i of those integers k in E that satisfy $n_{i+1} < k < n_i$. Denote by N_i the number of integers in E_i and by I the largest index form which N_i is non-zero.



Let us make two observations. First, if an integer k belongs to E, then so is $n - k$ since $f(n - k) = k$ (if $n - k$ can be reached by applying f and g, then so is k by applying f to $n-k$). Moreover, k belongs to E_0 if and only if $n-k$ does not. Indeed, it holds that $n = 2n_1$ or $n = 2n_1 + 1$. Therefore, if $k > n_1$, then

$$n - k < n - n_1 \leq 2n_1 + 1 - n_1 = n_1 + 1,$$

which means that $n - k \leq n_1$ and it cannot be in E_0. Vice versa, if k is in E but not in E_0 ($k < n_1$), then $n - k$ belongs to E_0 because

$$n - k > n - n_1 \geq 2n_1 - n_1 = n_1.$$

Hence, we can conclude that E_0 contains exactly as many elements as the rest of the exceptional sets. That is,

$$N_0 = N_1 + N_2 + \cdots + N_I.$$

The second observation is the following: if an integer k is an element of E_i and $i > 0$, then $2k$ and $2k + 1$ are elements of E_{i-1}. Indeed, they must belongs to the exceptional set E because if at least one them could be reached by applying functions f and g to n, then applying the function g once more, we would reach k and k would not be in the exceptional set. Moreover, they must be precisely in E_{i-1} because

$$n_i - 1 \leq 2n_{i+1} < 2k < 2k + 1 < 2n_i + 1 \leq n_{i-1} + 1$$

and hence $n_i \leq 2k < 2k + 1 \leq n_{i-1}$. The second observations means that if E_i contains an element k, then E_{i-1} contains at least two elements, namely $2k$ and $2k + 1$. Moreover, if E_i

contains two elements, say $k > \ell$, then E_{i-1} contains at least four elements, $2k+1 > 2k > 2\ell+1 > 2\ell$, and so on. Therefore, it is true that

$$N_{i-1} \geq 2N_i \quad \text{for all} \quad 1 \leq i \leq I.$$

Then

$$\begin{aligned} N_0 &\geq 2N_1 = N_1 + N_1 \geq N_1 + 2N_2 \geq N_1 + N_2 + 2N_3 \\ &\cdots \\ &\geq \big(N_1 + \cdots + N_{I-1} + N_I\big) + N_I \\ &> \big(N_1 + \cdots + N_{I-1} + N_I\big) = N_0, \end{aligned}$$

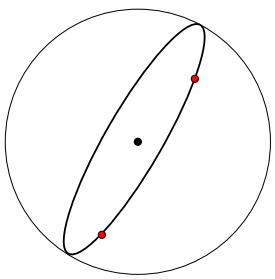
where the inequality on the last line is strict because $N_I > 0$ and the last equality is the consequence of the first observation. Since the statement $N_0 > N_0$ is absurd, the assumption that the set E is non-empty is false.

4. Problem 4

Let P be a convex potato floating in 3-dimensional space (convex means that the line segment between any two points of P lies within P). We are told that no matter how P is rotated, the points directly below P always form a perfectly round shadow. With this information show that P is a perfectly round potato.

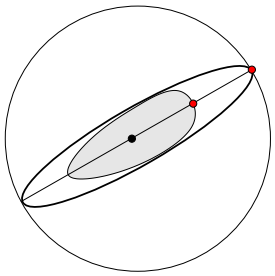
Put P within a perfectly round potato, that is, inside of a sphere. Shrink this sphere until it touches P in at least 2 distinct points. Lets call this smallest sphere S.

Through any two distinct points on a surface of a sphere one can draw an equator, a circle on the surface the sphere that separates it into two equal halves. To do this, rotate the sphere so that one of the points is the north pole and cut the sphere with a vertical plane that passes through the north pole and the other point.



Pick any 2 distinct points at which S touches P and draw the equator through them on the surface of S. Rotate S and P together so that the equator you draw is perfectly horizontal. Then the shadow of the sphere is a disk whose border is a circle, say C, which is the shadow of the equator. Moreover, the shadow of P is also a disk whose border is a circle that lies within C and touches C in at least 2 points. The latter is possible if and only if this is the same circle. Hence, since the shadows of P and S are always the same and coincide for at least one rotation, S and P have the same shadow all the time.

Let us show that not only the shadows but P and S coincide. Assume to the contrary that there is a point inside of S which is not in P. Draw a diameter through the center of S and this point. Since P is convex, when this diameter gets out of P (and it does because it goes through a point not in P) it stays outside of P. Hence, one of the points where the diameter touches the surface of S, say p, is not in P. Rotate S and P so that this diameter is perfectly



horizontal. Then there are no points of P on top, below, or at p. Therefore, the shadow of P does not cover the shadow of p, but the shadows must coincide. This contradiction shows that our assumption that S and P do not coincide was false and those are one and the same potato.

5. Problem 5

You have ten Scrabble© tiles with the letters A E F G I L M N O S *in that order. Your task is to shuffle them around to obtain a ten-letter English word (there is only one). However, you are only allowed two types of manipulation: you can shift the first letter to the last position, and you can pick the letters in odd positions (in order) followed by the letters in even positions (in order). Thus, starting with the letters in alphabetic order, you can reach* E F G I L M N O S A*, and* A F I M O E G L N S*. What is the shortest sequence of moves needed to complete the task?*

The only word that can be formed with the ten letters AEFGILMNOS is FLAMINGOES. Let's see first how to find *any* sequence of moves that reaches this word, and worry later about how to find the shortest one.

To begin, we need symbols for the two allowed moves. Let us write **f** for the one that sends the **f**irst letter to the right, and **s** for the one that **s**huffles even and odd letters. The first thing to notice is that **s** leaves the first and last letters untouched. Moreover, applying **s** three times reverses the order of the eight middle letters: $\mathbf{s}^3(1\ 23456789\ 0) = 1\ 98765432\ 0$. What happens if we alternate **f** and this reversal?

Playing a bit we find that **f** followed by a reversal, followed by $\mathbf{f}^9$ (which brings the last letter *back* to first position), followed by another reversal leaves unchanged the order of most letters: $\mathbf{fs}^3\mathbf{f}^9\mathbf{s}^3(123\ 4567890) = 1\ 4567890\ 23$. Then, another $\mathbf{f}^8$ brings the second and third letters to the left resulting in

$$\mathbf{fs}^3\mathbf{f}^9\mathbf{s}^3\mathbf{f}^8(123\ 4567890) = 231\ 4567890$$

The total effect is to shift the first three letters among themselves and leave the rest untouched. Let us call this sequence of moves $\mathbf{w} = \mathbf{f\ sss\ fffffffff\ sss\ fffffff}$ (so 24 moves in all). The advantage of using **w** is that, unlike **s**, the move **w** will not scramble letters after we have managed to put them in the correct order. Now we can use **f** to cycle the letters, and **w** to rearrange *selected* letters:

AEFGILMNOS	$\xrightarrow{\mathbf{f}^9}$	MGNOESFAIL	$\xrightarrow{\mathbf{w}^2}$
SAEFGILMNO	$\xrightarrow{\mathbf{w}^2}$	NMGOESFAIL	$\xrightarrow{\mathbf{f}^9}$
ESAFGILMNO	$\xrightarrow{\mathbf{f}^2}$	LNMGOESFAI	$\xrightarrow{\mathbf{w}^2}$
AFGILMNOES	$\xrightarrow{\mathbf{w}}$	MLNGOESFAI	$\xrightarrow{\mathbf{f}^9}$
FGAILMNOES	$\xrightarrow{\mathbf{f}}$	IMLNGOESFA	$\xrightarrow{\mathbf{w}}$
GAILMNOESF	$\xrightarrow{\mathbf{w}}$	MLINGOESFA	$\xrightarrow{\mathbf{f}^9}$
AIGLMNOESF	$\xrightarrow{\mathbf{f}^2}$	AMLINGOESF	$\xrightarrow{\mathbf{w}^2}$
GLMNOESFAI	$\xrightarrow{\mathbf{w}}$	LAMINGOESF	$\xrightarrow{\mathbf{f}^9}$
LMGNOESFAI	$\xrightarrow{\mathbf{f}}$	FLAMINGOES	✓

This is a total of 339 moves. Worse, it is very wasteful. Notice that the first ten moves are all **f** (because **w** begins with **f**), so they brought us back to the starting position! Surely we can do better...

A famous result in Algebra tells us that there is no general method to solve problems like this, other than plain trial and error. Since **fs** is not the same as **sf**, we have to search through all combinations of these two moves, and the possibilities grow out of control. We need a computer program. The one below is in pseuo-code, meaning that it in not actual code, but you can see how to translate it to your favorite computer language. The comments explain how each procedure works:

```
global int N

procedure{BinaryString}(int i)     % Input: integer i.
{                                  % Output: string of length N representing i in binary
 string binary                     %         (may need extra zeros to satisfy the length condition).

 binary = ""                       % Empty string.
 while(i > 0)                      % Concatenate a '0' or '1' depending on
    binary = binary + (i MOD 2)    % whether i is even or odd.
    i = i/2
 while(length(binary) < N)         % Add extra '0's if needed to get length N.
    binary = binary + '0'
 return(binary)
}

procedure{FirstRight}(string s)    % Input: string s.
{                                  % Output: same string after applying move 1.
 string shuffled

 shuffled[1] = s[2]
 shuffled[2] = s[3]
 shuffled[3] = s[4]
 shuffled[4] = s[5]
 shuffled[5] = s[6]
 shuffled[6] = s[7]
 shuffled[7] = s[8]
 shuffled[8] = s[9]
 shuffled[9] = s[10]
 shuffled[10] = s[1]
 return(shuffled)
}

procedure{Shuffle}(string s)       % Input: string s.
{                                  % Output: same string after applying move 2.
 string shuffled

 shuffled[1] = s[1]
 shuffled[2] = s[3]
 shuffled[3] = s[5]
```

```
 shuffled[4] = s[7]
 shuffled[5] = s[9]
 shuffled[6] = s[2]
 shuffled[7] = s[4]
 shuffled[8] = s[6]
 shuffled[9] = s[8]
 shuffled[10] = s[10]
 return(shuffled)
}

MAIN(int N)                        % As i takes all values, its binary
{                                  % representation describes all possible
 int i, j                          % combinations of moves f and s with
 string s, moves                   % length N.

 for i from 0 to 2^N-1
    s = "AEFGILMNOS"
    moves = BinaryString(i)
    for j from 1 to N
       if(moves[j] = '0') s = FirtRight(s)
       if(moves[j] = '1') s = Shuffle(s)
    if(s = "FLAMINGOES") print(i)
}
```

Running this program with input `N = 17` results in no answer, but with input `N = 18` we find one (and only one!) solution:

AEFGILMNOS	$\xrightarrow{\mathbf{f}}$	SEIMOAGLNF	$\xrightarrow{\mathbf{f}}$
EFGILMNOSA	$\xrightarrow{\mathbf{f}}$	EIMOAGLNFS	$\xrightarrow{\mathbf{f}}$
FGILMNOSAE	$\xrightarrow{\mathbf{s}}$	IMOAGLNFSE	$\xrightarrow{\mathbf{s}}$
FIMOAGLNSE	$\xrightarrow{\mathbf{s}}$	IOGNSMALFE	$\xrightarrow{\mathbf{s}}$
FMALSIOGNE	$\xrightarrow{\mathbf{s}}$	IGSAFONMLE	$\xrightarrow{\mathbf{s}}$
FASONMLIGE	$\xrightarrow{\mathbf{s}}$	ISFNLGAOME	$\xrightarrow{\mathbf{f}}$
FSNLGAOMIE	$\xrightarrow{\mathbf{f}}$	SFNLGAOMEI	$\xrightarrow{\mathbf{f}}$
SNLGAOMIEF	$\xrightarrow{\mathbf{s}}$	FNLGAOMEIS	$\xrightarrow{\mathbf{s}}$
SLAMENGOIF	$\xrightarrow{\mathbf{s}}$		
SAEGILMNOF	$\xrightarrow{\mathbf{s}}$	FLAMINGOES	✓