Go down deep enough into anything and you will find mathematics.
– **Dean Schlicler**

If you don't believe mathematics is simple, it is only because you don't realize how complicated life is. – **John von Neumann**

To ask the right question is harder than to answer it. – **Georg Cantor**

Learning is a treasure that follows its owner. – **Chinese Proverb**

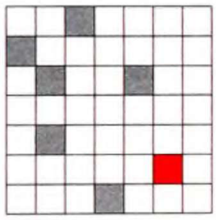
Chapter 1

A Taste of Discrete Mathematics

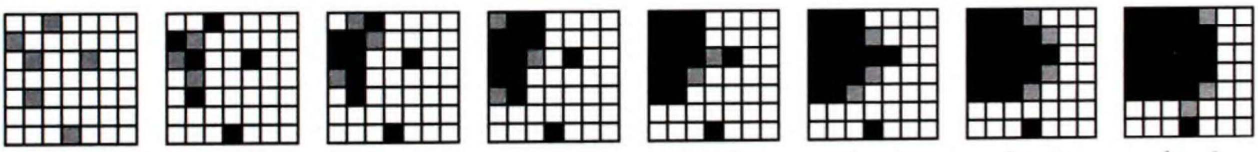
1: Wheting the appetite: epidemics; speed-dating; and, friendship networks.

Discrete mathematics deals with "objects that we count." When you cook your favorite Thai dish, you perform "discrete" steps in a sequence. We can count those steps. When delegates sit at a banquet, each "discrete" delegate sits at a single "discrete" seat. Processes that take place in well defined steps, involving objects that come in indivisible units are the focus of discrete mathematics. The most famous discrete object is the digital computer which executes instructions in sequential steps and uses a discrete scratch-paper called RAM to store intermediate results during its calculations. If that is not enough reason to study the subject, then for beauty's sake do it. Let us begin with some examples of discrete mathematics in the real world.

1.1 Modeling Epidemics



Outbreaks of a deadly virus EBOLA recur every now and then. The name of the virus is not important. People live on a grid, think of a chess board with only white squares. Each grid square is a person. Adjacent squares that share a side are neighbors. Initially, some people are infected, in gray. Infection is permanent and if at least two of your neighbors are infected today, then tomorrow you will be infected. Who will ultimately get infected? For example, will the square shaded red eventually get infected on the 7×7 grid shown? Let us tinker. Tinkering is an essential part of discrete mathematics. So, starting from the initial gray infections, let's see how the infection spreads over a few days. On each day, the previous infections are in black and the new infections are in gray.



Pop Quiz 1.1
Continue to tinker with the epidemic spread above. Does the red square eventually get infected?

The epidemic spread is indeed a discrete process. People are either infected or not. Time is discretized into days. We identified the actors as people living on a grid and the process dynamics – at least two of your neighbors must be infected to infect you. These are modeling assumptions. Modeling assumptions are critical, and you may argue with them. People may not live on a grid. Perhaps one neighbor being infected suffices to infect you. A good model is more likely to give correct conclusions. The beginning of a discrete mathematical analysis is always a model of the phenomenon you are analyzing. In our case, the model was a 2-contact threshold for epidemic spread on a grid. We do not address modeling, which is very application dependent. We care about what happens next. You ask questions. Here are a few interesting questions.

1. Given the initial infection, who will ultimately get infected?
2. What is the fewest initial infections needed to ultimately infect the whole community?
3. If you had a few vaccines, who should you immunize to minimize the ultimate infection?
4. Given the current observed state of the epidemic, can one determine the "points of entry", defined as the smallest set of initially infected people that could have produced the observed infections?

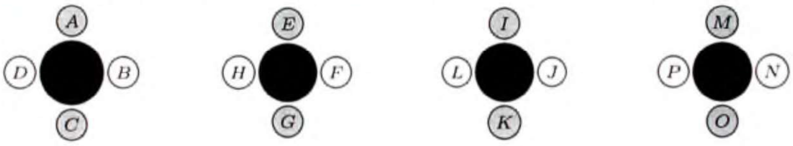
To get answers, one must analyze the model, and that is where discrete mathematics enters.

Exercise 1.2
Can you infect the entire 7×7 grid, starting with an initial infection of just 6 people?

Before we switch gears, observe that our model for EBOLA spread can apply to other contexts: virus spread in a computer network; company-defaults in an economic crisis; adoption of a technology in a social network.

1.2 Speed Dating

You analyze disease spread in your spare time, but your real job is to run a speed-dating club. Every night you get 16 people, 8 boys and 8 girls. Anyone can date any other person. Here is how the night plays out. You have four tables, and there are 4 rounds of speed dating. In each round, 4 people sit at each table to "speed date" in a group setting. Round 1 of speed dating is shown below. The letters are the first initial of the clients. In round 1, A meets B, C and D; B meets A, C and D; and so on.



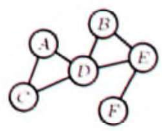
You succeed if you hook up many couples, so you want everyone to meet as many people as possible during the night. That's the model. Here are some interesting questions.

1. What does "...meet as many people as possible ..." mean? Do you care about the average number of encounters your clients had or the client who had the fewest encounters?
2. Can one efficiently configure the rounds so that everyone meets at least 10 people?
3. What would happen if you randomly assigned clients to tables in each round?

Tinker a little. Can you come up with good ways to configure the 4 rounds of speed-dating. See if you can figure out why I am so confident that no matter how much you tinker, no one will meet more than 12 people. I can also say that if you configured the rounds randomly, everyone would expect to meet about 9 people, so random is not that bad. You cannot beat 12 for anyone, and random gets you to about 9 for everyone.

1.3 Friendship Networks and Ads

Six people Alice (A), Bob (B), Charles (C), David (D), Edward (E) and Fiona (F) form a friendship or social network. Circles are the people and the links are the friendships between pairs of people. Two people, i.e. circles, are friends if they are connected by a line. You might recognize the friendship cliques between $\{A, C, D\}$ and $\{D, B, E\}$ while F looks like a loner.[1]



We can visually analyze our small friendship network, but what about an online social network with a billion people? We certainly can't draw it on this page. So how would you go about identifying all the large friendship cliques? This turns out to be a very interesting challenge, but why do we care?

[1] In a friendship clique, everyone is friends with everyone-else.

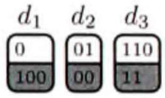
An advertiser who wants to market a new smartphone may try to convince David (D) to adopt the device in the hopes that everyone in David's friendship cliques might also buy the device. That would be a huge payoff. To make such advertising dreams into realities, we need to model social networks in a way that can be represented on a computer, and find all the large friendship-cliques or social "communities" so the advertiser can identify whom to target. Finding the large cliques is a tough discrete mathematics problem.

1.4 Modeling Computers

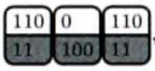
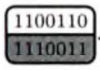
You've now seen some flavors of discrete mathematics. Computer scientists use discrete mathematics to model, analyze and solve real world problems. The summit of our adventure is going to be a grand model, a model of the digital computer – a *model of computing*. We want a realistic model that captures your desktop as well as smartphone, GPU or fitbit. But, it should be simple enough to analyze, for we have deep questions to ask.

1. What can we compute?
2. What *can't* we compute?
3. Are there things we can compute in principle, but it takes too long?

In answering these questions, we will journey through the world of discrete mathematics.



What is computing? Let's get a feel for it using a domino puzzle. The top and bottom entry in each of the three dominos d_1, d_2, d_3 on the right is a binary string. A sequence of dominoes produces a combined domino in which the top string is the concatenation of all the top strings in order, and similarly for the bottom string. For example,

$d_3 d_1 d_3 =$ , which gives the combined domino .

To solve the domino puzzle, find a sequence of dominos for which the combined top and bottom strings match. Repetition of dominos is allowed and you need not use all dominos. In this case, you can verify that $d_3 d_2 d_3 d_1$ solves the puzzle. That's nice, but what does a simple kids' puzzle have to do with computing? Could you write a program to solve the domino puzzle? Your program would read in a text file where each row of the file describes a domino, two comma separated binary strings. Your program should output a sequence of dominos which solves the puzzle, or say it can't be done. Now, does that look like a computing problem to you, something that could be on a programming assignment?

> **Challenge.** I'm feeling sly and evil. A prize of **$1,000** inflation adjusted goes to the first correct program that solves the domino puzzle for any input file of dominos. There are two catches.
> (i) Your program must *stop* and output the correct answer no matter what the input domino file.
> (ii) You must give a *proof* that your program is correct.

We can't rely on intuition to say what is and is not computing. We need a precise model. Stay tuned.

1.5 Proof

It is Human to seek proof, to ask why. If your neighbor says evacuate because of hurricanes, wouldn't you seek verification? If you claim to have a perpetual motion machine, I won't take you at your word. We each have different thresholds to be convinced of something. In life, a few true instances are often enough – that is inductive proof. The sun has risen every morning. That's enough for us earthy beings to conclude that the sun will always rise. Logically, a few cases of the sun having risen does not prove "The sun rises every morning." It only means the statement is not obviously false. In mathematics, we have the same urge for verification, but our expectations are high. We require deductive proof.

In the speed-dating ritual on page 8, nobody meets more than 12 people. Here is a proof. In any round a person meets at most 3 new people. So, after 4 rounds they meet at most $4 \times 3 = 12$ new people. Aren't you utterly convinced? The beauty of deductive proof is that it leaves no room for doubt.

Becoming Good at Discrete Math. You're no stranger to reasoning in real life. You can look at a rental contract and determine when you can break the lease without penalty. Mathematics, reasoning about abstract objects, is similar yet seems difficult. Don't fret. You had a lifetime of training for everyday reasoning. With similar training, you can build stamina for mathematics. But you must be diligent. Mathematics is no spectator sport. Don't "read" or "study". *Do*! Work the examples, exercises and problems. Write in the book. Annotate definitions and theorems with pictures or simple examples. Make sure you understand what is being said. No one speed reads mathematics. Even to those fluent in the language, it is a foreign tongue.

In mathematics, if you're missing something, you're missing everything. *Work* the text, quizzes and exercises, with pencil and paper in hand. You must agree with the smallest detail.

It is worth showcasing the recurring workflow an expert uses to solve a problem. Memorize it.

1: Model the problem you are trying to solve using a discrete mathematical object.
2: Tinker with easy cases to understand the model. **Tinkering is essential.**
3: Based on the tinkering, formulate a conjecture about your problem/model.
4: Prove the conjecture and make it a theorem. You now *know* something new.

The novice builds the model and stares at it without knowing what to prove, because of a failure to tinker.

Exercise 1.3 [A Sisyphean Puzzle to Teethe On]

Zeus punished King Sisyphus of Corinth to an eternity of rolling a boulder up a hill only to see it roll down when near the top. Sisyphean tasks are laborious yet futile.

Three boxes start with 100, 200 and 300 boulders, the configuration $(100, 200, 300)$. A move places a stone from one box into another. For example, moving a stone from the box with 200 to the box with 100 gives configuration $(101, 199, 300)$. Each move gains or loses gold coins and debt is allowed. The gain is larger for a move from a box with many stones to a box with few stones. Specifically,

gain = # stones left in originating box (after move) − # stones in destination box (before move).

Here is a sequence of moves with the corresponding payments.

start configuration	end configuration	gain
$(\mathbf{100}, 200, 300)$	$(101, \mathbf{199}, 300)$	$199 - 100 = +99$
$(101, 199, \mathbf{300})$	$(101, \mathbf{198}, 301)$	$198 - 300 = -102$
$(\mathbf{101}, 198, 301)$	$(102, 198, \mathbf{300})$	$300 - 101 = +199$

The profit for these three moves is $+196$ coins $(= 99 - 102 + 199)$. Sisyphus can move stones, but must return to the start configuration. How much can Sisyphus profit? Can you prove it?

Mathematics and "Partial Credit". They don't mesh well. In school, partial credit is a learning-aide. It's comforting to know you're close to a solution, and partial credit delivers the message. Mathematics is not so forgiving. A proof is right or wrong. There's no almost proven. Computer programs are mathematical objects running on mathematical devices. A program works or it doesn't. If a builder makes one wrong join the whole structure can fall. You don't get $\frac{1}{2}$-credit for sending someone to the moon but forgetting to bring them back. A program that mostly works is a catastrophe waiting to happen. Check out Therac-25 on Wikipedia:

"Because of concurrent programming errors, it sometimes gave its patients radiation doses that were hundreds of times greater than normal, resulting in death or serious injury."

Who cares if Therac-25 had one bug or 17 bugs? Should we be lenient because Therac-25 worked 99.9% of the time, only failing in rare boundary cases? There is a fundamental difference between algorithms that always work and heuristics which often work, but without guarantee. Life does not give credit for partial solutions.

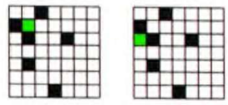
Critical computer systems from traffic control to robotic surgery to self-driving cars must fully work all the time, otherwise people will suffer. Take the precaution to *prove* your program works. Others rely on it.

1.6 Problems

Problem 1.1. The parity of an integer is 0 if it is even and 1 if it is odd. Which operations preserve parity:
(a) Multiplying by an even. (b) Multiplying by an odd. (c) Raising to a positive integer power.

Problem 1.2. What's wrong with this comparison: Google's nett worth in 2017, about \$700 billion, exceeds the GDP of many countries, e.g. Argentina's 2016-GDP was about \$550 billion. (Look up nett worth and GDP.)

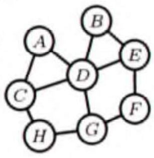
Problem 1.3. Consider 2-contact EBOLA on a grid. You have one immunization vaccine. We show two different immunization scenarios, where you immunize the green square. Show the final infection in each case and determine which person you prefer to immunize? How many vaccines are needed to ensure that nobody else gets infected?



Problem 1.4. For the speed-dating problem with 16 people, $A, B, \ldots, P$ and four tables, arrange the rounds so that:
(a) In two rounds, everyone meets 6 people. (c) In four rounds, everyone meets 12 people.
(b) In three rounds, everyone meets 9 people. (d) In five rounds, everyone meets 15 people?

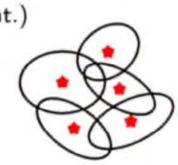
Problem 1.5 (Social Golfer Problem). 32 golfers form 8 groups of 4 each week. Each group plays a round of golf. No two golfers can be in the same group more than once. For how many weeks can this golfing activity go on?
(a) "Prove" that this golfing activity cannot go on for more than 10 weeks.
(b) Try to create a scheduling of players for as many weeks as you can. (10 is possible.)
(c) How is this problem related to the speed-dating problem?
In general you must schedule g groups of golfers each of size s for w weeks so that no two golfers meet more than once in the same group. Given (g, s, w), can it can be done and what is the schedule? This is a hard problem.

Problem 1.6. Students $A, \ldots, H$ form a friendship network (right). To advertise a new smartphone, you plan to give some students free samples. Here are two models for the spread of phone-adoption.
Model 1 (WEAK MAJORITY): People buy a phone if at least as many friends have the phone as don't.
Model 2 (STRONG MAJORITY): People buy a phone if more friends have the phone than don't



(a) Use your intuition and determine the most "central" of the people in this friend-network.
(b) If you give a phone only to this central node, who ultimately has a phone in: (i) Model 1 (ii) Model 2?
(c) How many phones must you distribute, and to whom, so that everyone switches to your phone in Model 2?
(d) Repeat part (c), but now you cannot give a phone to the central node.
(A slight change to a model can have a drastic impact on the conclusions. A good model is important.)

Problem 1.7. Five radio stations (red stars) broadcast to different regions, as shown. The FCC assigns radio-frequencies to stations. Two radio stations with overlapping broadcast regions must use different radio-frequencies so that the common listners do not hear garbled nonsense.
What is the minimum number of radio-frequencies the government needs?



Discrete math problems are like childhood puzzles. Parity, symmetry and invariance often yield simple solutions.

Problem 1.8. Two players take turns placing identical circular quarters on a circular table. Coins cannot overlap and must remain on the table. The last person to play wins. Do you want to go first or second? *[Hint: symmetry.]*

Problem 1.9. A chocolate-bar has 50 squares (5×10). How many breaks are necessary to break the bar into its 50 individual squares? You may only break a piece along a straight line from one side to the other. No stacking allowed. *[Hint: Define the invariant $\Delta = \#pieces - \#breaks$. What happens to Δ with each break?]*

Problem 1.10. A single-elimination tournament has 57 players. Players may receive byes in some rounds. How many matches are played before a winner is declared? Does it depend on how the tournament is configured? *[Hint: Define the invariant $\Delta = \#players\ remaining + \#matches\ played$. What happens to Δ after a match?]*

Problem 1.11. Five pirates must share 100 gold coins. The most senior pirate proposes a division of coins and all pirates vote. If at least half the pirates agree, the coins are divided as proposed. If not, the proposer is killed and the process continues with the next most senior pirate. A pirates priority is to stay alive, and then to get as much gold as possible. What should the senior pirate propose? *[Hint: Sometimes it is better to start with a smaller problem.]*

Problem 1.12. Can you color squares of a 9×9 grid blue or red so that every square has one opposite color neighbor (neighbors are left, right, up or down).

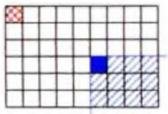
Problem 1.13. 57 security guards are positioned so that no two pairs of guards are the same distance apart. Every guard watches the guard closest to him. Is there an arrangement of the guards so that every guard is being watched?

Problem 1.14. 10 trucks each have 100 gallons of fuel and use 1 gallon of fuel per mile. How far can you deliver a chest that fits in one truck? (You can transfer the chest and/or fuel from truck to truck.)

Problem 1.15. A camel owner wants to sell his 300 bananas at a market 100 miles away. The camel can carry at most 100 bananas, but eats a banana for every mile travelled. How many bananas can be sold at the market?

Problem 1.16. Show that fewer than n initial infections cannot infect the whole $n \times n$ grid in 2-contact EBOLA. *[Hint: For a square, define 4-outgoing links (N,S,E,W) to its 4 neighbors. Pretend boundary-squares have neighbors. For an infected square, remove all outgoing links to infected neighbors. Let Δ, the "wavefront" of the infected area, be all remaining outgoing links for infected squares. Can Δ increase? What is Δ when all squares are infected?]*

Problem 1.17 (Chomp). In the grid of chocolate, if you eat the top-left square, you lose. Each player takes turns to eat a square plus all the chocolate below it and to the right. We show a possible first move and the chocolate that removed in blue. Do you want to go first or second? *[Hint: Either eating the bottom right piece wins or not. If not, what should you do?]*

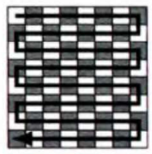


Problem 1.18. A man has a boat which can carry him and one other thing. How can the man get a fox, a chicken and a bag of corn across the river, if, when unattended, the fox eats the chicken and the chicken eats the corn.

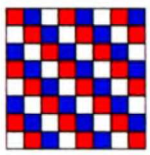
Problem 1.19. Tasks involving covering an area using different shaped tiles are a treasure trove of interesting puzzles.

(a) Remove the top left and bottom right squares on an 8×8 chess board. Can you tile the remaining 62 squares with 31 dominos? *[Hint: Show that #black squares − #white squares is an <u>invariant</u> when you place a domino?]*

(b) On an 8×8 chess board, show that if you remove any two squares of different colors, you can tile the remainder of the board with dominos. *[Hint: See illustration on the right. We show a path starting from the top-left. You can tile the board by placing dominos along this path. You may assume that the first square removed is white (why?). Show that you can still tile the remaining board along the path.]*



(c) On a 8×8 chess board, show that if you remove any corner square, you cannot tile the remainder of the board with straight triominos ($\square\square\square$). It is possible to tile the board with triominos after removing one square. Can you identify which squares can be removed (there are 4)? *[Hint: See illustration on the right. We have colored the squares on the chess board so that a triomino must cover one square of each color.]*



(d) Can you cover a 10×10 chessboard with 25 straight tetrominos ($\square\square\square\square$). If yes, how?

Problem 1.20. There are 13 purple, 15 red and 17 green chameleons. When chameleons of different colors meet they both transform to the third color. Will all 45 chameleons ever be the same color? *[Hint: Consider Δ = #purple − #red.]*

Problem 1.21. A building has 1000 floors. You wish to determine the highest floor from which you can drop an egg without the egg breaking. If you had 1000 identical eggs, you could drop one from each floor and see which eggs survive. How many egg drop trials do you need if you have: (a) One egg. (b) Two identical eggs.

Problem 1.22. Four boys take 1min, 2min, 7min and 10min to cross a bridge. The bridge only holds two boys at a time. It is dark and there is only one flashlight, which is needed to cross the bridge. Two boys cross at the speed of the slower boy who holds the flashlight. All four boys must get across the bridge. If the fastest boy acts as chauffeur for the other three, all four can cross in 21 min. Can all four get across the bridge faster?

Problem 1.23. Two consecutive positive numbers n and $n+1$ are given to you and a friend. One player gets n and the other gets $n+1$, at random. You look at your number and shout out your opponents number if you know it, otherwise you pass the turn to your opponent. Will this game ever stop?

Problem 1.24. You have a gold chain with 63 links. You would like to cut some links to obtain a set of links of different sizes. You goal is to be able to represent any number of links from 1 to 63 as a collection of some of your pieces in order to trade. What is the minimum number of links you need to cut to be able to do so?

Problem 1.25. Three ants a, b, c are on the vertices A, B, C of a triangle. Each ant randomly picks one of the other vertices and walks to it. What are the chances that no ants colide on an edge or at a destination vertex)?
What if there are four ants a, b, c, d on the vertices A, B, C, D of a tetrahedron?

Problem 1.26. Two players alternately pick numbers without replacement from the set $\{1, 2, 3, \ldots, 9\}$. The first player to obtain three numbers that sum to 15 wins. What is your strategy?

Problem 1.27. A maharaja has 100 amphoras of wine. A traitor poisons one amphora, gets detected and killed. The poisoned amphora is not known and the poison kills in exactly one month. The maharaja uses tasters to tell if wine is safe, depending on whether the taster lives or dies after a month.

(a) The maharaja wants to safely drink wine in a month, what is the minimum number of tasters he needs.

(b) The maharaja wants to use all safe amphoras to throw an orgy in a month. What is the minimum number of tasters he needs. A simple solution is 100 tasters, one on each amphora. One can do much better though.

Problem 1.28. Two players take turns picking a coin from either end of a line of 20 coins. In the example below, if player 1 always takes from the left and player 2 from the right, then player 1's coins total 80, and player 2's total is 146.

3 7 4 19 13 2 14 7 5 6 7 11 8 2 32 47 11 6 9 13

The player with the highest total wins, player 2 in the example. Do you want to play first or second?

Problem 1.29. To weigh sugar, you have a comparison scale that can compare weights (illustrated). Give the fewest weights that are needed to measure out $1, 2, \ldots, 121$ pounds of sugar.

For example, to measure 3 pounds of sugar with 2 and 5 pound weights, place the sugar and 2 pounds on the one side, and 5 pounds on the other side. The sugar weighs 3 pounds if the scale balances.

Problem 1.30. More than half of 99 processors are good and the rest are bad. You may ask a processor to evaluate another processor. A good processor always gives the correct answer and a bad one gives the wrong answer. How many times must you ask some (any) processor to evaluate another before you can identify a good processor?

Problem 1.31. A plane has fuel capacity to fly half way around the world. A plane can refuel from another plane in mid-air. All planes are at the airport. How many planes and tanks of gas do you need so that you can support a single plane to fly around the world? All planes must return to the airport.

Problem 1.32. 25 horses have different speeds. You can race up to 5 horses at a time and observe the order in which the horses finish. You have no stop-watch. Show that 7 races suffice to determine the fastest 3 horses.

Problem 1.33. 100 prisoners are up for a pardon. Prisoners will be lined in random order with a randomly chosen red or blue hat on each head. A prisoner sees only those ahead of them in the line. The last in line shouts the color of his hat. If he gets it right, he is pardoned. Then the second-last prisoner gets a chance and so on until the first in line.

The night before pardoning, the prisoners may strategize. During the pardoning process, the prisoners cannot communicate except to shout out a hat color. If the prisoners optimally strategize the night before, what are the chances that the first to shout is pardoned, the second to shout, the third to shout and so on up to the final prisoner?

Problem 1.34. At a puzzle-party with 32 guests, the host will shuffle a 52-card deck and paste a card on each guest's forhead. A guest will see every other guest's card but not their own card. After the cards are pasted on forheads, each guest, one by one, must shout out a card (e.g. 4♠). At the end the number of guests who correctly shouted out their card is multiplied by \$1,000 to get a prize amount which is split evenly among all guests.

Intense discussion breaks out among the guests as they arrive. A philosopher suggests breaking into 16 pairs. In each pair, the first to shout says their partner's card so the partner can guess correctly. This strategy guarantees \$16,000. A FOCS-student claims, "I can guarantee we will share \$31,000." Can you come up with a strategy to guarantee \$31,000?

Problem 1.35. Three friends A, B, C each have tokens a, b, c. At every step a random pair of friends is picked to swap whatever tokens they currently have. If the first pair picked is (A, B) and then (A, C) then the tokens are distributed c, a, b after the two swaps. What are the chances each friend has their own token after 2015 swaps?

Problem 1.36. On a table are some red and blue cards. Two players take turns picking two cards. If the two cards picked are the same color, both cards are replaced by one red card. If the two cards picked are different colors, both cards are replaced by one blue card. When one card remains, you win if it is blue and your opponent wins if it is red.

(a) Must the game always end, or can it go on forever?

(b) Who wins if there are 8 blue and 11 red cards to start? Does it matter who goes first? *[Hint: Parity invariant.]*

Problem 1.37. Dad normally picks Sue from school which ends at 3pm. School ended early at 2pm, so Sue started walking home and dad picked her up on the way, returning home 20min earlier than usual. For how long did Sue walk?

Problem 1.38. Pick any six kids. Show that either 3 of them know each other or 3 of them do not know each other.

Problem 1.39. Fifteen houses are in a row. A thief robs a house. On each subsequent night, the thief robs a neighbor of the house robbed the previous night. The thief may backtrack and rob the same house. A policeman can watch any one house per night. Is there a strategy for the policeman to guarantee catching the thief?

Problem 1.40. 5 of 10 coins are showing heads. You can move coins to form two sets, and you can flip over any coins you wish. How will you guarantee that both sets have the same number of heads showing, blindfolded?

Problem 1.41. Baniaz and her twin kids pass a gumball machine with 2 red, 3 blue and 4 green gumballs. Gumballs cost 1¢ each and come out randomly. Baniaz buys gumballs until she can give each of her kids one gumball of the same color. In the worst case, how much must Baniaz be willing to spend? What if she had quadruplets instead?

Problem 1.42. Two 1 meter fuses (strings) each burn non-uniformly in 60 sec. How can you measure 45 sec?

Here come hard problems that take you to the boundaries of mathematics and computing.

Problem 1.43 (Collatz/$3n+1$ Problem). Given an integer $n > 1$, repeat as follows until you reach 1:

$$n \longrightarrow \begin{cases} n/2 & \text{if } n \text{ is even;} \\ 3n+1 & \text{if } n \text{ is odd;} \end{cases}$$

Example: $6 \to 3 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$. Do you reach 1 for every n? This "simple" problem is unsolved!

Problem 1.44 (Subset Sum). Find two different subsets of this set of one hundred 27-digit numbers, with the requirement that the numbers in each subset must have the same sum.

1:	5719825393567961346558155629	35:	8794353172213177612939776215	69:	7549684656732941456945632221
2:	5487945882843158696672157984	36:	2989694245827479769152313629	70:	2397876675349971994958579984
3:	4767766531754254874224257763	37:	6117454427987751131467589412	71:	4675844257857378792991889317
4:	1855924359757732125866239784	38:	2761854485919763568442339436	72:	2832515241382937498614676246
5:	4289776424589197647513647977	39:	6884214746997985976433695787	73:	8755442772953263299368382378
6:	7967131961768854889594217186	40:	8671829218381757417536862814	74:	9833662825734624455736638328
7:	2572967277666133789225764888	41:	9431156837244768326468938597	75:	5298671253425423454611152788
8:	1294587141921952639693619381	42:	4788448664674885883585184169	76:	9857512879181186421823417538
9:	4764413635323911361699183586	43:	3624757247737414772711372622	77:	1471226144331341144787865593
10:	1474343641823476922667154474	44:	9361819764286243182121963365	78:	3545439374321661651385735599
11:	2578649763684913163429325833	45:	9893315516156422581529354454	79:	6735367616915626462272211264
12:	5161596985226568681977938754	46:	5913625989853975289562158982	80:	2141665754145475249654938214
13:	2242632698981685551523361879	47:	8313891548569672814692858479	81:	8481747257332513758286947416
14:	7474189614567412367516833398	48:	2265865138518379114874613969	82:	9961217236253576952797397966
15:	6211855673345949471748161445	49:	3477184288963424358211752214	83:	9941237996445827218665222824
16:	4942716233498772219251848674	50:	6321349612522496241515883378	84:	6242177493463484861915865966
17:	5516264359672753836539861178	51:	1796439694824213266958886393	85:	4344843511782912875843632652
18:	5854762719618549417768925747	52:	6366252531759955676944496585	86:	7568842562748136518615117797
19:	5313691171963952518124735471	53:	8545458545636898974365938274	87:	2776621559882146125114473423
20:	6737691754241231469753717635	54:	3362291186211522318566852576	88:	6174299197447843873145457215
21:	4292388614454146728246198812	55:	8464473866375474967347772855	89:	5387584131525787615617563371
22:	4468463715866746258976552344	56:	2892857564355262219965984217	90:	5317693353372572284588242963
23:	2638621731822362373162811879	57:	4296693937661266715382241936	91:	6612142515552593663955966562
24:	1258922263729296589785418839	58:	8634764617265724716389775433	92:	1314928587713292493616625427
25:	4482279727264797827654899397	59:	8415234243182787534123894858	93:	2446827667287451685939173534
26:	8749855322285371162986411895	60:	2267353254454872616182242154	94:	9786693878731984534924558138
27:	1116599457961971796683936952	61:	4689911847578741473186337883	95:	2926718838742634774778713813
28:	3879213273596322735993329751	62:	4428766787964834371794565542	96:	3791426274497596641969142899
29:	9212359131574159657168196759	63:	7146295186764167268433238125	97:	2831727715176299968774951996
30:	3351223183818712673691977472	64:	2273823813572968577469388278	98:	3281287353463725292271916883
31:	8855835322812512868896449976	65:	6686132721336864457635223349	99:	9954744594922386766735519674
32:	4332859486871255922555418653	66:	3161518296576488158997146221	100:	3414339143545324298853248718
33:	2428751582371964453381751663	67:	1917611425739928285147758625		
34:	6738481866868951787884276161	68:	3516431537343387135357237754		

Problem 1.45 (Verifier for "Hello World"). Write a program in your pet language to solve the this problem.

Input: Any C^{++} program F.cpp (an ASCII text file).

Output: *Yes* if: when you compile and run F.cpp, it prints "Hello World", and eventually stops.

No if: when you compile and run F.cpp, the program loops forever or stops without printing "Hello World".

Would you have guessed that a solver for the domino puzzle (Section 1.4) can be used to build a Hello-World-verifier?