Chapter 5

Problemas ACM

5.1 Tree Reconstruction

You have just finished a compiler design homework question where you had to find the parse tree of an expression. Unfortunately you left your assignment in the library, but luckily your friend picked it up for you. Instead of e-mailing you the parse tree so that you can rewrite the solution, your friend decides to play a practical joke and sends you just the DFS and BFS trace. Rather than try to redo the entire question you decide to reconstruct the tree.

5.2 Network

A Telephone Line Company (TLC) is establishing a new telephone cable network. They are connecting several places numbered by integers from 1 to N. No two places have the same number. The lines are bidirectional and always connect together two places and in each place the lines end in a telephone exchange. There is one telephone exchange in each place. From each place it is possible to reach through lines every other place, however it need not be a direct connection, it can go through several exchanges. From time to time the power supply fails at a place and then the exchange does not operate. The officials from TLC realized that in such a case it can happen that besides the fact that the place with the failure is unreachable, this can also cause that some other places cannot connect to each other. In such a case we will say the place (where the failure occured) is critical. Now the officials are trying to write a program for finding the number of all such critical places. Help them.

5.3 Il Gioco dell'X

The game 'Il Gioco dell' X' is played on a N by N board ($N > 1$). The object of both players, say Black and White, is to join opposite sides of the board by placing in turn their pawns on the board in such a way that a path is made from one side to the other by adjacent (neighboring) pawns of their own color. Although the board is NxN it is not a square but rather diamond-shaped. Let us denote the field on the board in row i and column j by $(i, j)(1 \leq i, j \leq N)$. The neighbors of (i, j) are:

```
( i-1 , j-1 ) , ( i-1 ,  j  )
(  i  , j-1 ) , (  i  , j+1 )
( i+1 ,  j  ) , ( i+l , j+1 )
```

provided these fields do not fall outside the board.

Black tries to join row 1 with row N, while White tries to join column 1 with column N.

lt is a Deep Mathematical Result that this game cannot end in a draw (that is, without winner). As we will present to you only full boards, there will always be a winner. It may, however, be difficult to see who has won, so some computer assistance would be appreciated.

5.3.1 Example

Example 1 Example 2

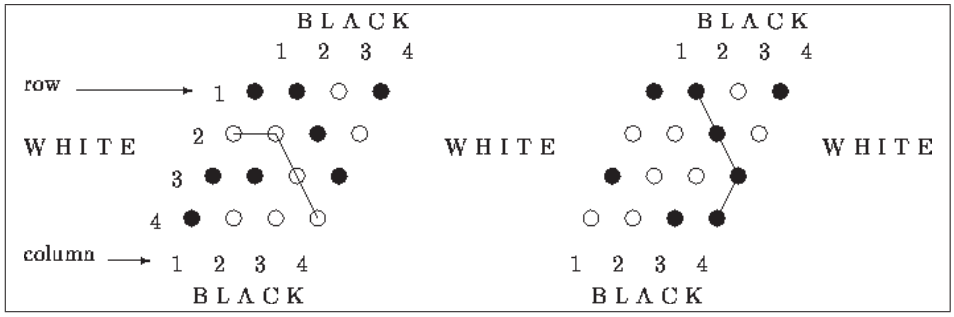

Figure 5.1: Ejemplos

In example 1 White has won, and in example 2 Black has won.

5.4 Numbering Paths

Problems that process input and generate a simple "yes" or "no" answer are called decision problems. One class of decision problems, the NP-complete problems, are not amenable to general efficient solutions. Other problems may be simple as decision problems, but enumerating all possible "yes" answers may be very difficult (or at least time-consuming).

This problem involves determining the number of routes available to an emergency vehicle operating in a city of one-way streets.

5.4.1 Problem

Given the intersections connected by one-way streets in a city, you are to write a program that determines the number of different routes between each intersection. A route is a sequence of one-way streets connecting two intersections.

Intersections are identified by non-negative integers. A one-way street is specified by a pair of intersections. For example, (j, k) indicates that there is a one-way street from intersection j to intersection k. Note that two-way streets can be modeled by specifying two one-way streets: *(j, k)* y *(k, j)*.

Consider a city of four intersections connected by the following one-way streets:

```
0  1
0  2
1  2
2  3
```

There is one route from intersection 0 to 1, two routes from 0 to 2 (the routes are $0 \to 1 \to 2$ y $0 \to 2$), two routes from 0 to 3, one route from 1 to 2, one route from 1 to 3, one route from 2 to 3, and no other routes.

It is possible for an infinite number of different routes to exist. For example if the intersections above are augmented by the street *(3, 2)* , there is still only one route from 0 to 1, but there are infinitely many different routes from 0 to 2. This is because the street from 2 to 3 and back to 2 can be repeated yielding a different sequence of streets and hence a different route. Thus the route $0 \to 2 \to 3 \to 2 \to 3 \to 2$ is a different route than $0 \to 2 \to 3 \to 2$.

5.5 Exchange Rates

Using money to pay for goods and services usually makes life easier, but sometimes people prefer to trade items directly without any money changing hands. In order to ensure a consistent "price", traders set an exchange rate between items. The exchange rate between two items A and B is expressed as two positive integers m and n, and says that m of item A is worth n of item B. For example, 2 stoves might be worth 3 refrigerators. (Mathematically, 1 stove is worth 1.5 refrigerators, but since it's hard to find half a refrigerator, exchange rates are always expressed using integers.)

Your job is to write a program that, given a list of exchange rates, calculates the exchange rate between any two items.

The input file contains one or more commands, followed by a line beginning with a period that signals the end of the file. Each command is on a line by itself and is either an assertion or a query. An assertion begins with an exclamation point and has the format

! m itema = n itemb

where itema and itemb are distinct item names and m and n are both positive integers less than 100. This command says that m of itema are worth n of itemb. A query begins with a question mark, is of the form

? itema = itemb

and asks for the exchange rate between itema and itemb, where itema and itemb are distinct items that have both appeared in previous assertions (although not necessarily the same assertion). For each query, output the exchange rate between itema and itemb based on all the assertions made up to that point. Exchange rates must be in integers and must be reduced to lowest terms. If no exchange rate can be determined at that point, use question marks instead of integers. Format all output exactly as shown in the example.

Note:

- Item names will have length at most 20 and will contain only lowercase letters.
- Only the singular form of an item name will be used (no plurals).
- There will be at most 60 distinct items.
- There will be at most one assertion for any pair of distinct items.
- There will be no contradictory assertions. For example, "2 pig = 1 cow", "2 cow = 1 horse", and "2 horse = 3 pig" are contradictory.

- Assertions are not necessarily in lowest terms, but output must be.
- Although assertions use numbers less than 100, queries may result in larger numbers that will not exceed 10000 when reduced to lowest terms.

5.6 Hanoi Tower Troubles Again!

People stopped moving discs from peg to peg after they know the number of steps needed to complete the entire task. But on the other hand, they didn't not stopped thinking about similar puzzles with the Hanoi Tower. Mr.S invented a little game on it. The game consists of N pegs and a LOT of balls. The balls are numbered 1,2,3... The balls look ordinary, but they are actually magic. If the sum of the numbers on two balls is NOT a square number, they will push each other with a great force when they're too closed, so they can NEVER be put together touching each other.

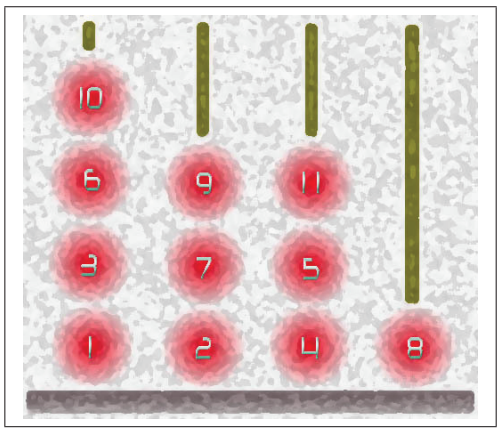

Figure 5.2: Variación de Hanoi

The player should place one ball on the top of a peg at a time. He should first try ball 1, then ball 2, then ball 3... If he fails to do so, the game ends. Help the player to place as many balls as possible. You may take a look at the picture above, since it shows us a best result for 4 pegs.

5.7 Rank the Languages

You might have noticed that English and Spanish are spoken in many areas all over the world. Now it would be nice to rank all languages according to the number of states where they are spoken.

5.7.1 Problem

You're given a map which shows the states and the languages where they are spoken. Look at the following map:

```
ttuuttdd
ttuuttdd
uuttuudd
uuttuudd
```

The map is read like this: Every letter stands for a language and states are defined as connected areas with the same letter. Two letters are "connected" if one is at left, at right, above or below the other one. So in the above map, there are three states where the language "t" is spoken, three where "u" is spoken and one state where people speak "d".

Your job is to determine the number of states for each language and print the results in decreasing order.

5.8 Best Roots

Tree is an important data structure. Searching is a basic operation in any data structure. In a tree searching mainly depends on its height. Consider the following three trees.

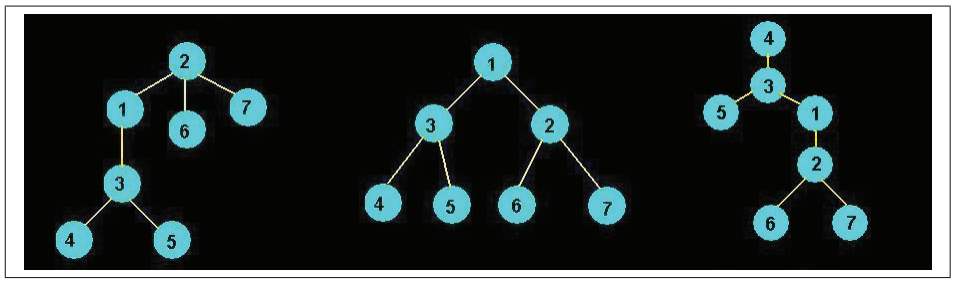

Figure 5.3: Mejores Raices

If you observe carefully, you will see that all trees are same except different nodes are used as roots. Here the height of the tree varies with the selection of the root. In the 1st tree root is '2' and height is 3. In 2nd one root is '1' and height is 2. And in last one root is '4' and height is 4. We will call '1' best root as it keeps the tree with the least possible height and '4' worst root for the opposite reason.

In this problem, you have to find out all best roots and worst roots for a given tree.

5.9 Trust groups

The personnel department of Association of Cookie Monsters (ACM) has noticed that the productivity of various work groups in the company is not as good as it could be. They have interviewed the employees in the affected groups and they have detected the root of the problem: trust (or, rather, the lack thereof). Some employees do not trust the rest of the group, and this is decreasing their motivation and happiness. The personnel department wants to solve this problem, and has decided to reorganize the groups so that they are stable, i.e., they are formed by people who trust each other. They have asked the employees, and they know the people each employee trusts directly. Moreover, if employee A trusts employee B and employee B trusts employee C, then employee A will trust employee C. And obviously, each employee trusts himself. They want to create as few groups as possible to reduce administration overhead (they also do not want to work too hard).

With this information they have contacted you, and asked you to write a program that finds the minimum number of stable groups that can be created.

5.10 Instant View of Big Bang

Have you forgot about wormholes? Oh my god! Ok, let me explain again.

A wormhole is a subspace tunnel through space and time connecting two star systems. Wormholes have a few peculiar properties:

1. Wormholes are one-way only.
2. The time it takes to travel through a wormhole is negligible.
3. A wormhole has two end points, each situated in a star system.
4. A star system may have more than one wormhole end point within its boundaries.
5. Between any pair of star systems, there is at most one wormhole in each direction.

6. There are no wormholes with both end points in the same star system.

All wormholes have a constant time difference between their end points. For example, a specific wormhole may cause the person traveling through it to end up 15 years in the future. Another wormhole may cause the person to end up 42 years in the past.

A brilliant physicist, wants to use wormholes to study the Big Bang. Since warp drive has not been invented yet, it is not possible for her to travel from one star system to another one directly. This can be done using wormholes, of course.

The scientist can start her journey from any star system. Then she wants to reach a cycle of wormholes somewhere in the universe that causes her to end up in the past. By traveling along this cycle a lot of times, the scientist is able to go back as far in time as necessary to reach the beginning of the universe and see the Big Bang with her own eyes. Write a program to help her to find such star systems where she can start her journey.

5.11 A Node Too Far

To avoid the potential problem of network messages (packets) looping around forever inside a network, each message includes a Time To Live (TTL) field. This field contains the number of nodes (stations, computers, etc.) that can retransmit the message, forwarding it along toward its destination, before the message is unceremoniously dropped. Each time a station receives a message it decrements the TTL field by 1. If the destination of the message is the current station, then the TTL field's value is ignored. However, if the message must be forwarded, and the decremented TTL field contains zero, then the message is not forwarded.

In this problem you are given the description of a number of networks, and for each network you are asked to determine the number of nodes that are not reachable given an initial node and TTL field value. Consider the following example network:

If a message with a TTL field of 2 was sent from node 35 it could reach nodes 15, 10, 55, 50, 40, 20 and 60. It could not reach nodes 30, 47, 25, 45 or 65, since the TTL field would have been set to zero on arrival of the message at nodes 10, 20, 50 and 60. If we increase the TTL field's initial value to 3, starting from node 35 a message could reach all except node 45.

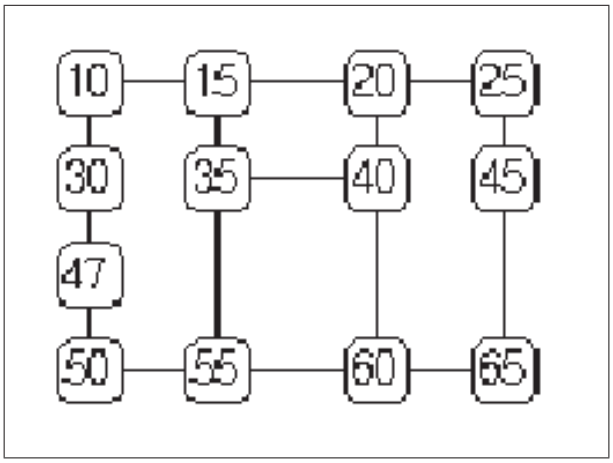

Figure 5.4: Nodos de la red

5.12 Wetlands of Florida

A construction company owns a large piece of real estate within the state of Florida. Recently, the company decided to develop this property. Upon inspection of the property, however, it was revealed that the land, at various locations, contained bodies of water. This came as a shock to the owners of the company, for they were from out of state and not familiar with wetlands of Florida. The situation was very grave and the owners not knowing that such bodies of water can be converted to beautiful lakes that will increase the value of the land around them, were about to abandon the construction project. Fortunately, this fact was brought to the owners' attention by a smart FIU graduate who worked for the company and consequently the construction project started.

The engineers divided the construction site by a grid into uniform square cells such that each square cell entirely contained either water or land. (How they did it, of course, is anybody's guess.) Now, the question that the engineers are to answer is the following: "Given the row and column number of a grid cell that contains water, what is the area of the lake containing that cell." (an area is measured by number of grid cells it contains. Diagonal cells are considered to be adjacent.)

You are to write a program to answer this question!

5.13 Risk

Risk is a board game in which several opposing players attempt to conquer the world. The gameboard consists of a world map broken up into hypothetical countries. During a player's turn, armies stationed in one country are only allowed to attack only countries with which they share a common border. Upon conquest of that country, the armies may move into the newly conquered country.

During the course of play, a player often engages in a sequence of conquests with the goal of transferring a large mass of armies from some starting country to a destination country. Typically, one chooses the intervening countries so as to minimize the total number of countries that need to be conquered. Given a description of the gameboard with 20 countries each with between 1 and 19 connections to other countries, your task is to write a function that takes a starting country and a destination country and computes the minimum number of countries that must be conquered to reach the destination. You do not need to output the sequence of countries, just the number of countries to be conquered including the destination. For example, if starting and destination countries are neighbors, then your program should return one.

The following connection diagram illustrates the first sample input

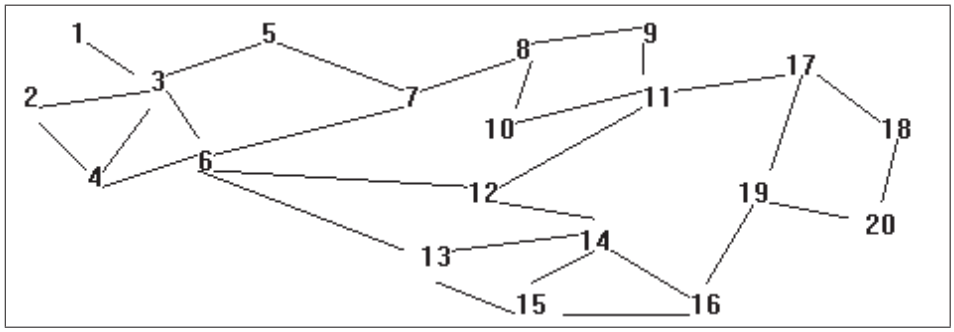

Figure 5.5: Risk: ejemplo de entrada

5.14 Oil Deposits

The GeoSurvComp geologic survey company is responsible for detecting underground oil deposits. GeoSurvComp works with one large rectangular region of land at a time, and creates a grid that divides the land into numerous square plots. It then analyzes each plot separately, using sensing equipment to determine whether or not the plot contains oil.

A plot containing oil is called a pocket. If two pockets are adjacent, then they are part of the same oil deposit. Oil deposits can be quite large and may contain numerous pockets. Your job is to determine how many different oil deposits are contained in a grid.

5.15 We Ship Cheap

The We-Ship-Cheap package shipping company is always interested in reducing their costs. They have decided that a computerized shipping route planner that determines the shortest shipping route between two cities would speed package delivery. We-Ship-Cheap services a number of different cities. They have established direct shipping links between pairs of cities. It is somewhat unusual that they have been able to create all the direct links with exactly the same distance. Unfortunately, since not every pair of cities is connected by a direct link, the shortest shipping route often involves travel through multiple intermediate cities.

Write a program that given a collection of cities and links between them, and a shipping request, prints out the shortest shipping route for the given shipping request.

5.16 Erdös Numbers

The Hungarian Paul Erdös (1913-1996, speak as "Ar-dish") not only was one of the strangest mathematicians of the 20th century, he was also one of the most famous. He kept on publishing widely circulated papers up to a very high age and every mathematician having the honor of being a co-author to Erdös is well respected.

Not everybody got the chance to co-author a paper with Erdös, so many people were content if they managed to publish a paper with somebody who had published a scientific paper with Erdös. This gave rise to the so-called Erdös numbers. An author who has jointly published with Erdös had Erdös number 1. An author who had not published with Erdös but with somebody with Erdös number 1 obtained Erdös number 2, and so on.

5.16.1 Problem

Today, nearly everybody wants to know which Erdös number he or she has. Your task is to write a program which computes Erdös numbers for a given set of scientists.

5.17 The Monocycle

A monocycle is a cycle that runs on one wheel and the one we will be considering is a bit more special. It has a solid wheel colored with five different colors as shown in the figure:

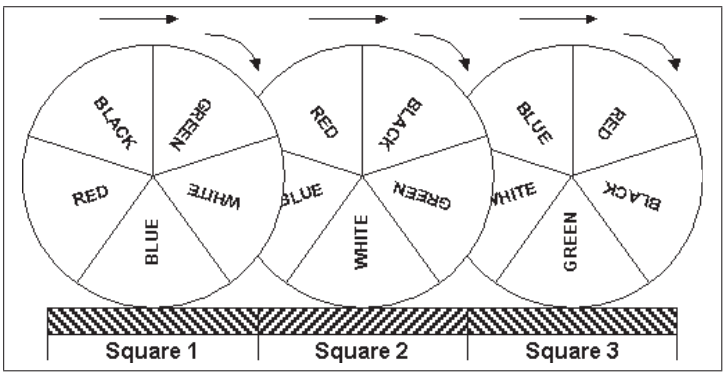

Figure 5.6: monocycle

The colored segments make equal angles (72^o) at the center. A monocyclist rides this cycle on an $M \times N$ grid of square tiles. The tiles have such size that moving forward from the center of one tile to that of the next one makes the wheel rotate exactly 72^o around its own center. The effect is shown in the above figure. When the wheel is at the center of square 1, the midŋpoint of the periphery of its blue segment is in touch with the ground. But when the wheel moves forward to the center of the next square (square 2) the midŋpoint of its white segment touches the ground.

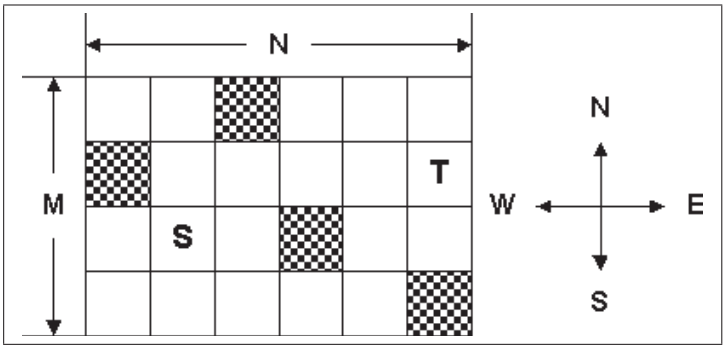

Figure 5.7: monocycle grid

Some of the squares of the grid are blocked and hence the cyclist cannot move

to them. The cyclist starts from some square and tries to move to a target square in minimum amount of time. From any square either he moves forward to the next square or he remains in the same square but turns 90^o left or right. Each of these actions requires exactly 1 second to execute. He always starts his ride facing north and with the midpoint of the green segment of his wheel touching the ground. In the target square, too, the green segment must be touching the ground but he does not care about the direction he will be facing.

Before he starts his ride, please help him find out whether the destination is reachable and if so the minimum amount of time he will require to reach it.

5.18 The Most Distant State

The 8-puzzle is a square tray in which eight square tiles are placed. The remaining ninth square is uncovered. Each tile has a number on it. A tile that is adjacent to the blank space can be slid into that space. A game consists of a starting state and a specified goal state. The starting state can be transformed into the goal state by sliding (moving) the tiles around. The 8-puzzle problem asks you to do the transformation in minimum number of moves.

```
2 8 3           1 2 3
1 6 4    =>     8   4
7   5           7 6 5
Start           Goal
```

However, our current problem is a bit different. In this problem, given an initial state of the puzzle your are asked to discover a goal state which is the most distant (in terms of number of moves) of all the states reachable from the given state.

5.19 Killing Aliens in Borg Maze

The Borg is an immensely powerful race of enhanced humanoids from the delta quadrant of the galaxy. The Borg collective is the term used to describe the group consciousness of the Borg civilization. Each Borg individual is linked to the collective by a sophisticated subspace network that insures each member is given constant supervision and guidance.

Your task is to help the Borg (yes, really) by developing a program which helps the Borg to estimate the minimal cost of scanning a maze for the assimilation of

aliens hiding in the maze, by moving in north, west, east, and south steps. The tricky thing is that the beginning of the search is conducted by a large group of over 100 individuals. Whenever an alien is assimilated, or at the beginning of the search, the group may split in two or more groups (but their consciousness is still collective.). The cost of searching a maze is defined as the total distance covered by all the groups involved in the search together. That is, if the original group walks five steps, then splits into two groups each walking three steps, the total distance is 11=5+3+3.

5.20 Roads in the North

Building and maintaining roads among communities in the far North is an expensive business. With this in mind, the roads are built in such a way that there is only one route from a village to a village that does not pass through some other village twice.

Given is an area in the far North comprising a number of villages and roads among them such that any village can be reached by road from any other village. Your job is to find the road distance between the two most remote villages in the area.

The area has up to 10,000 villages connected by road segments. The villages are numbered from 1.

5.21 Babel

John and Mary are brothers, and are enthusiastic about their courses on foreign languages. Each of the brothers is taking several language courses. When they get home they comment on grammar, vocabulary, culture of the different countries and so on. In one of those conversations they realized some words are common to more than one language, even though the words may have different meanings in the languages. For example, the word "amigo" exists in Portuguese and Spanish and has the same meaning, while "date" is a word that exists in English and French and may have different meanings, since "date" is also a fruit, besides meaning a calendar date. On the other hand, "red" in Spanish is a network, while in English it is a color.

Thrilled by these findings, the brothers decided to write in a notepad all words in common they could think of, associating each word to a pair of languages. Observant and smart, John proposed a challenge to Mary: given one language to start

and one language to finish, write down a sequence of words such that the first word is included in the vocabulary of the start language, and the last word is included in the vocabulary of the finish language. Two adjacent words in the sequence must be in the vocabulary of the same language. For example, if the start language is Portuguese and the finish language is French, Mary could write the sequence "amigo actual date" (Portuguese/Spanish, Spanish/English, English/French).

To John's surprise, Mary solved the problem rather easily. Annoyed by his sister's success, he decided to make the problem more difficult: Mary must find a solution in which the sequence has the smallest number of letters in total (not counting spaces between words), and, besides, two consecutive words must not have the same initial letter.

Note that the previous solution is now invalid, as "amigo" and "actual" share the same initial letter. It is possible, however, to find another solution, "amigo red date", with a total length equal to 12.

John did an extensive research on the Internet and compiled an enormous list of words, and challenged Mary to solve the problem. As there may be more than one solution, he asked her to answer if there is a solution, and in that case to answer the number of letters in the best solution. Can you help Mary?

5.22 Adventure of Super Mario

After rescuing the beautiful princess, Super Mario needs to find a way home – with the princess of course :-) He's very familiar with the 'Super Mario World', so he doesn't need a map, he only needs the best route in order to save time.

There are A Villages and B Castles in the world. Villages are numbered 1..A, and Castles are numbered A+1..A+B. Mario lives in Village 1, and the castle he starts from is numbered A+B. Also, there are two-way roads connecting them. Two places are connected by at most one road and a place never has a road connecting to itself. Mario has already measured the length of every road, but they don't want to walk all the time, since he walks one unit time for one unit distance(how slow!).

Luckily, in the Castle where he saved the princess, Mario found a magic boot. If he wears it, he can super-run from one place to another IN NO TIME. (Don't worry about the princess, Mario has found a way to take her with him when super-running, but he wouldn't tell you :-P)

Since there are traps in the Castles, Mario NEVER super-runs through a Castle. He always stops when there is a castle on the way. Also, he starts/stops super-runnings ONLY at Villages or Castles.

Unfortunately, the magic boot is too old, so he cannot use it to cover more than L kilometers at a time, and he cannot use more than K times in total. When he comes back home, he can have it repaired and make it usable again.

5.23 Knights in FEN

There are black and white knights on a 5 by 5 chessboard. There are twelve of each color, and there is one square that is empty. At any time, a knight can move into an empty square as long as it moves like a knight in normal chess (what else did you expect?).

Given an initial position of the board, the question is: what is the minimum number of moves in which we can reach the final position which is:

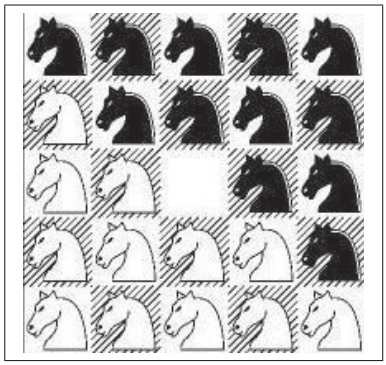

Figure 5.8: Movimiento de caballos

5.24 Dijkstra, Dijkstra.

You are a political prisoner in jail. Things are looking grim, but fortunately, your jailmate has come up with an escape plan. He has found a way for both of you to get out of the cell and run through the city to the train station, where you will leave the country. Your friend will escape first and run along the streets of the city to the train station. He will then call you from there on your cellphone (which somebody smuggled in to you inside a cake), and you will start to run to the same train station. When you meet your friend there, you will both board a train and be on your way to freedom.

Your friend will be running along the streets during the day, wearing his jail clothes, so people will notice. This is why you can not follow any of the same streets that your friend follows - the authorities may be waiting for you there. You have to pick a completely different path (although you may run across the same intersections as your friend).

What is the earliest time at which you and your friend can board a train?

5.25 Retorno a casa

Manuel y Antonio son amigos y les gusta ir a ver los juegos de beisbol en el estadio. Al terminar el juego deben regresar rápido a sus casas, pero les gusta mucho comentar sobre el partido durante el regreso, por lo que andan el mayor tiempo por el mismo camino.

Ellos viven en una ciudad que puede ser modelada como un conjunto de calles y uniones. Cada calle conecta a un par distinto de uniones y pueden ser caminadas en ambas direcciones. No hay dos calles que conecten a las mismas dos uniones. Ellos no viven juntos, ni viven en el estadio. Hay al menos un camino desde el estadio a la casa de Manuel y lo mismo ocurre con la casa de Antonio.

Dada la información acerca de las calles y las uniones en la ciudad, la ubicación del estadio, la casa de Manuel y la casa de Antonio, usted debe decirle a Manuel y a Antonio la distancia máxima que pueden caminar juntos sin obligarlos a caminar más de la mínima distancia desde el estadio a sus respectivas casas (ninguno caminara más de lo necesario para llegar a su casa, solo quieren hablar por el mayor tiempo posible). Ellos también desean saber cuánto tendrán que caminar cada uno solo.

5.26 Optimal Programs

As you know, writing programs is often far from being easy. Things become even harder if your programs have to be as fast as possible. And sometimes there is reason for them to be. Many large programs such as operating systems or databases have “bottlenecks” - segments of code that get executed over and over again, and make up for a large portion of the total running time. Here it usually pays to rewrite that code portion in assembly language, since even small gains in running time will matter a lot if the code is executed billions of times.

In this problem we will consider the task of automating the generation of optimal assembly code. Given a function (as a series of input/output pairs), you are to come up with the shortest assembly program that computes this function.

The programs you produce will have to run on a stack based machine, that supports only five commands: ADD, SUB, MUL, DIV and DUP. The first four commands pop the two top elements from the stack and push their sum, difference, product or integer quotient1 , respectively, on the stack. The DUP command pushes an additional copy of the top-most stack element on the stack.

So if the commands are applied to a stack with the two top elements a and b (shown to the left), the resulting stacks look as follows:

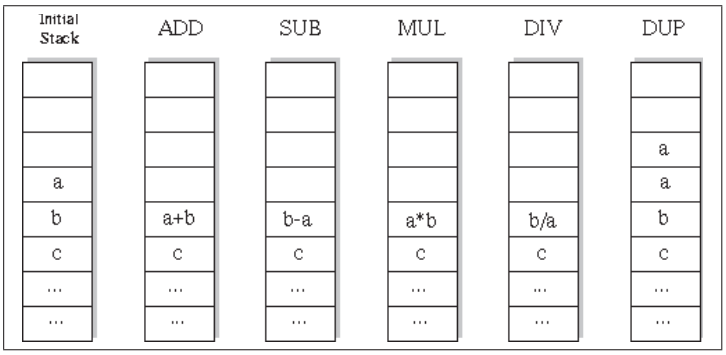

Figure 5.9: Ejemplos

At the beginning of the execution of a program, the stack will contain a single integer only: the input. At the end of the computation, the stack must also contain only one integer; this number is the result of the computation.

There are three cases in which the stack machine enters an error state:

- A DIV-command is executed, and the top-most element of the stack is 0.
- A ADD, SUB, MUL or DIV-command is executed when the stack contains only one element.
- An operation produces a value greater than 30000 in absolute value.

5.26.1 Input

The input consists of a series of function descriptions. Each description starts with a line containing a single integer n ($n \leq 10$), the number of input/output pairs to follow. The following two lines contains n integers each: $x_1, x_2, \ldots, x_n$ in the first line (all different), and $y_1, y_2, \ldots, y_n$ in the second line. The numbers will be no more than 30000 in absolute value.

The input is terminated by a test case starting with n = 0. This test case should not be processed.

```
4
1 2 3 4
0 -2 -6 -12
3
1 2 3
1 11 1998
1
1998
1998
0
```

5.26.2 Output

You are to find the shortest program that computes a function f , such that f(xi) = yi for all $i \in \{1, \ldots, n\}$. This implies that the program you output may not enter an error state if executed on the inputs xi (although it may enter an error state for other inputs). Consider only programs that have at most 10 statements.

For each function description, output first the number of the description. Then print out the se- quence of commands that make up the shortest program to compute the given function. If there is more than one such program, print the lexicographically smallest. If there is no program of at most 10 statements that computes the function, print the string "Impossible". If the shortest program consists of zero commands, print "Empty sequence".

Output a blank line after each test case.

```
Program 1
DUP DUP MUL SUB

Program 2
Impossible

Program 3
Empty sequence
```