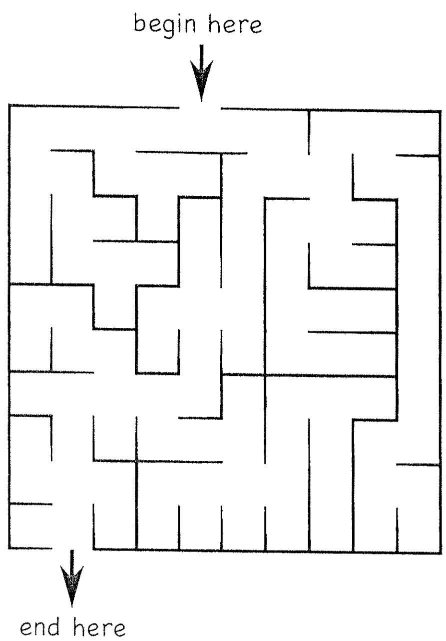
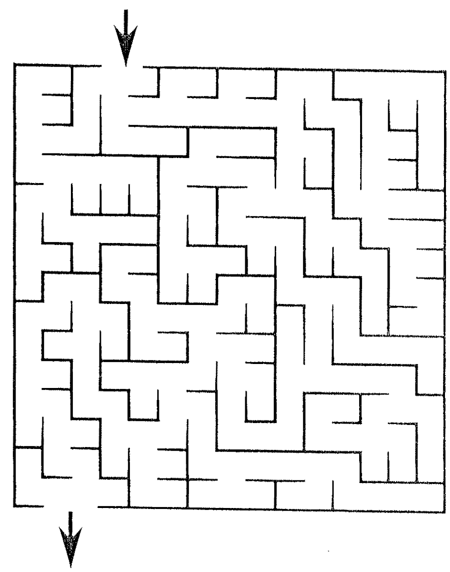
How to build great mazes

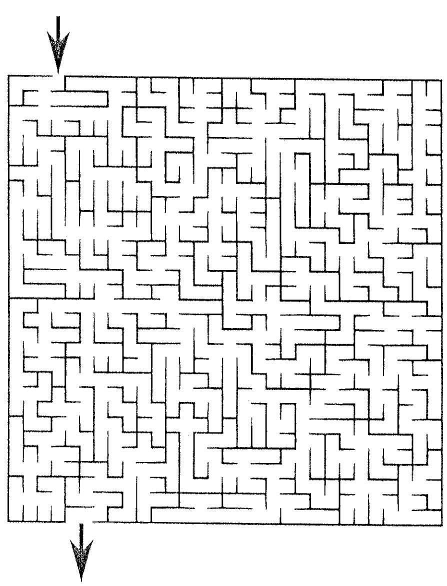
Renato Feres - Math Circle 2/5/2012

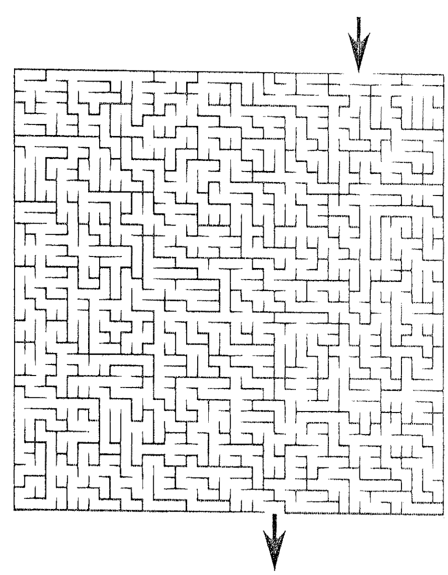
Warm-up: can you solve some of these maze problems?

(You don't have to solve all of them here!)



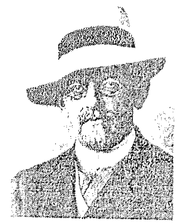




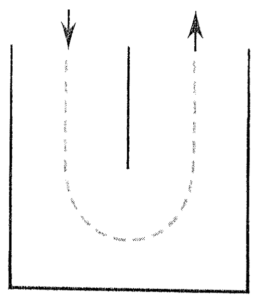


A curious example (Hilbert mazes)

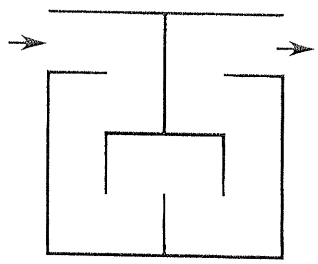
This is a sequence of mazes named after the famous German mathematician David Hilbert (1862-1943).



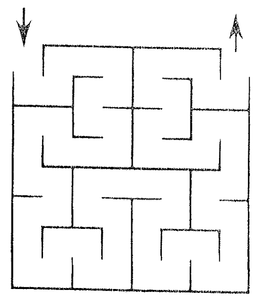
There is one maze for each "order" 1, 2, 3, ...



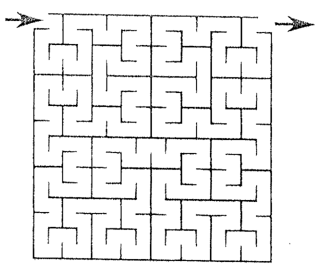
Hilbert maze of order 1
(The dashed line is a solution path.)



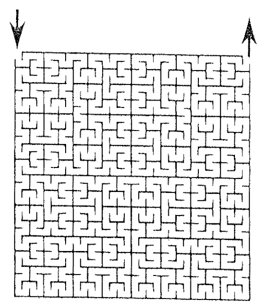
order 2



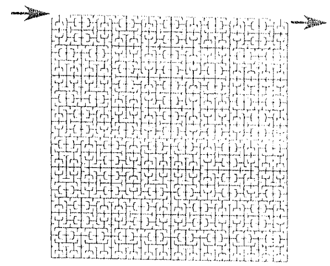
order 3



order 4

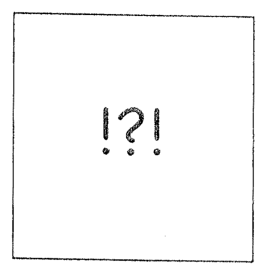


order 5



order 6

...



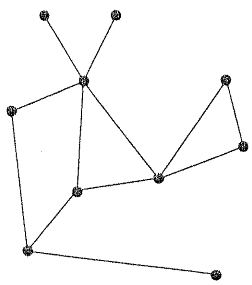
infinite order

What happens to the solution path as the order grows to infinity?

A recipe for building mazes

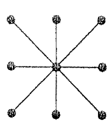
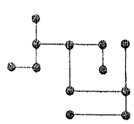
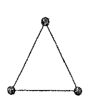
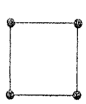
Rather than solving maze problems, what we would really like to do is find a way of building good mazes. For this purpose, it will be useful to learn a few ideas from the theory of **graphs**.

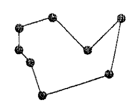
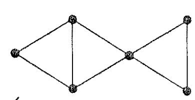
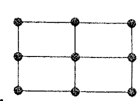
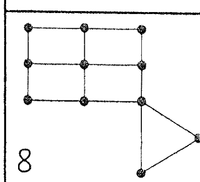
A **graph** (or network) consists of a collection of **vertices** (the dots) and a collection of **edges** (the connecting lines).



A graph with 10 vertices (V=10) and 12 edges (E=12). The number n=E-V+1=3 has an interesting interpretation. What is it?

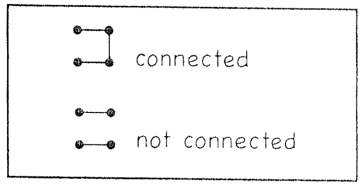
Exercise 1: For each of the following graphs, calculate the numbers V, E, n=E-V+1.

	E	V	n
 1			
 2			
 3			
 4			

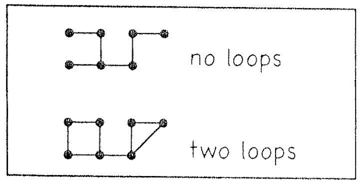
	E	V	n
 5			
 6			
 7			
 8			

Trees

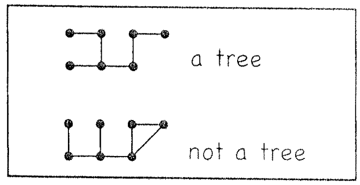
A graph is **connected** if every pair of vertices can be joined by a path that moves along the edges.



A **loop** in a graph is a path without backtracking that begins and ends at the same vertex.



A **tree** is a connected graph without any loops.

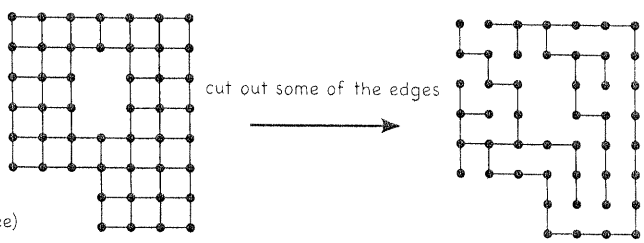


Exercise 2: Which of the graphs of exercise 1 are trees? What is the value of n when the graph is a tree? What is the relationship between n and the number of loops?

Spanning trees (in a grid)

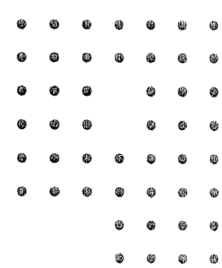
A **spanning tree** of a graph is a tree obtained by eliminating edges until there are no more loops, without splitting the graph into disconnected pieces. So, it is a tree inside the graph which still links all the vertices.

A graph (not a tree)



A spanning tree. Note that
1. all vertices are still linked,
2. but there are no loops.
If you put a single edge back a loop is created.

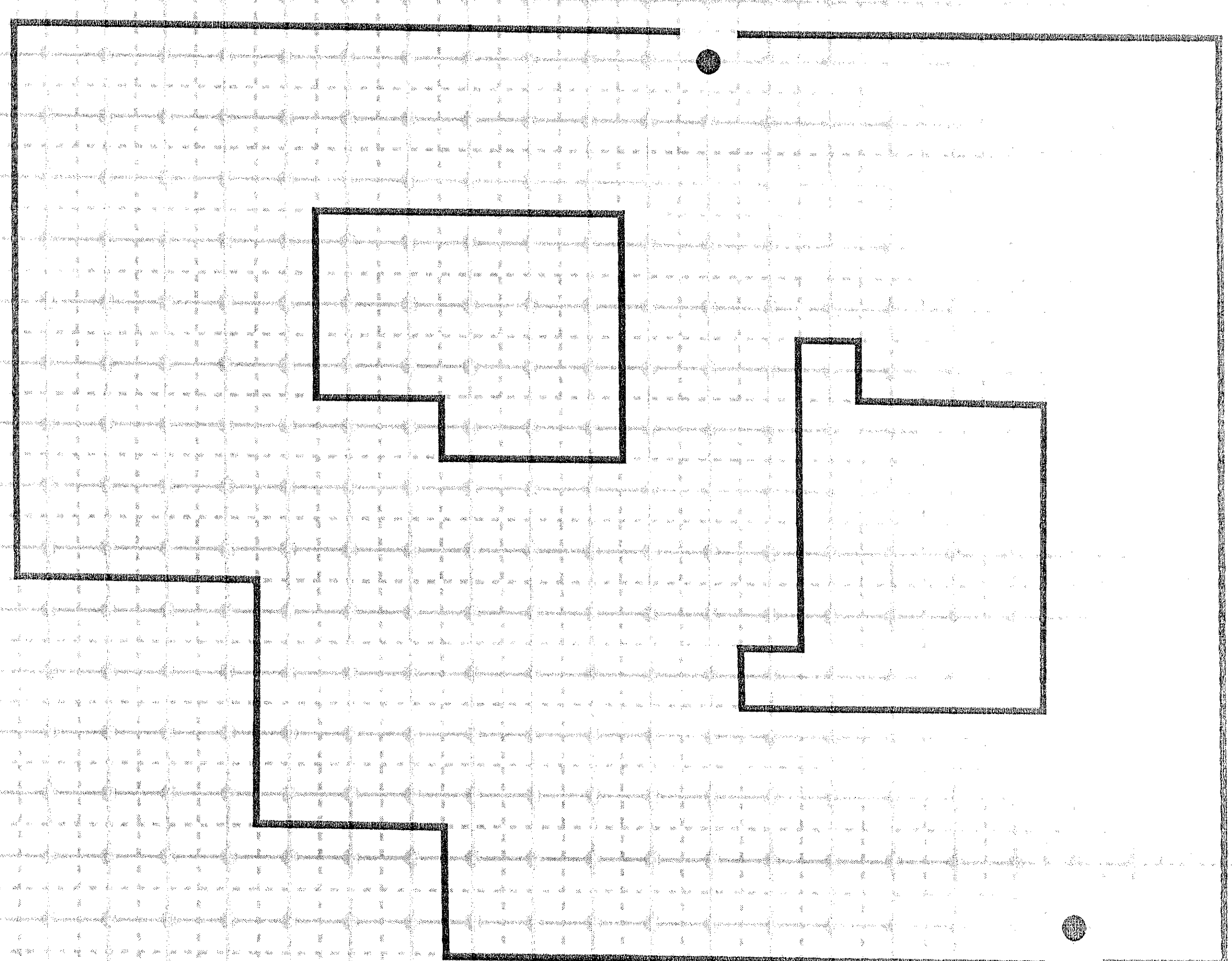
Exercise 3: Draw a different spanning tree for this same graph. Remember: Keep enough edges so that the vertices are all linked without forming loops. (Don't connect dots that are not linked in the initial graph.)



Now back to mazes

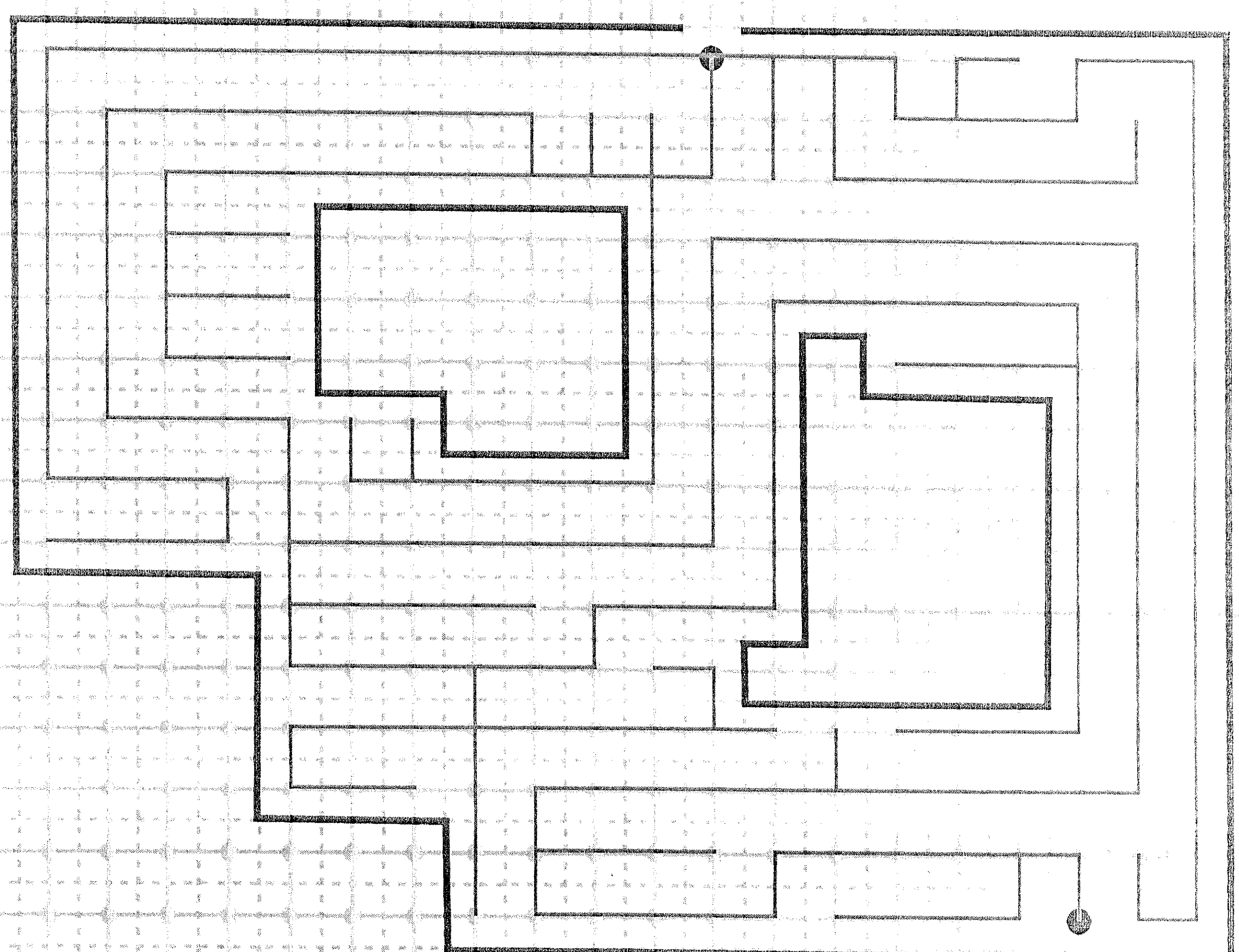
It is useful to work over a grid as the one shown here. The dots and solid lines will be used for tracing maze paths and the dashed lines for tracing the walls of the maze.

Step 1. With a pen, draw an enclosing wall with entrance and exit doors, moving along dashed lines. You may add interior walls if you wish.

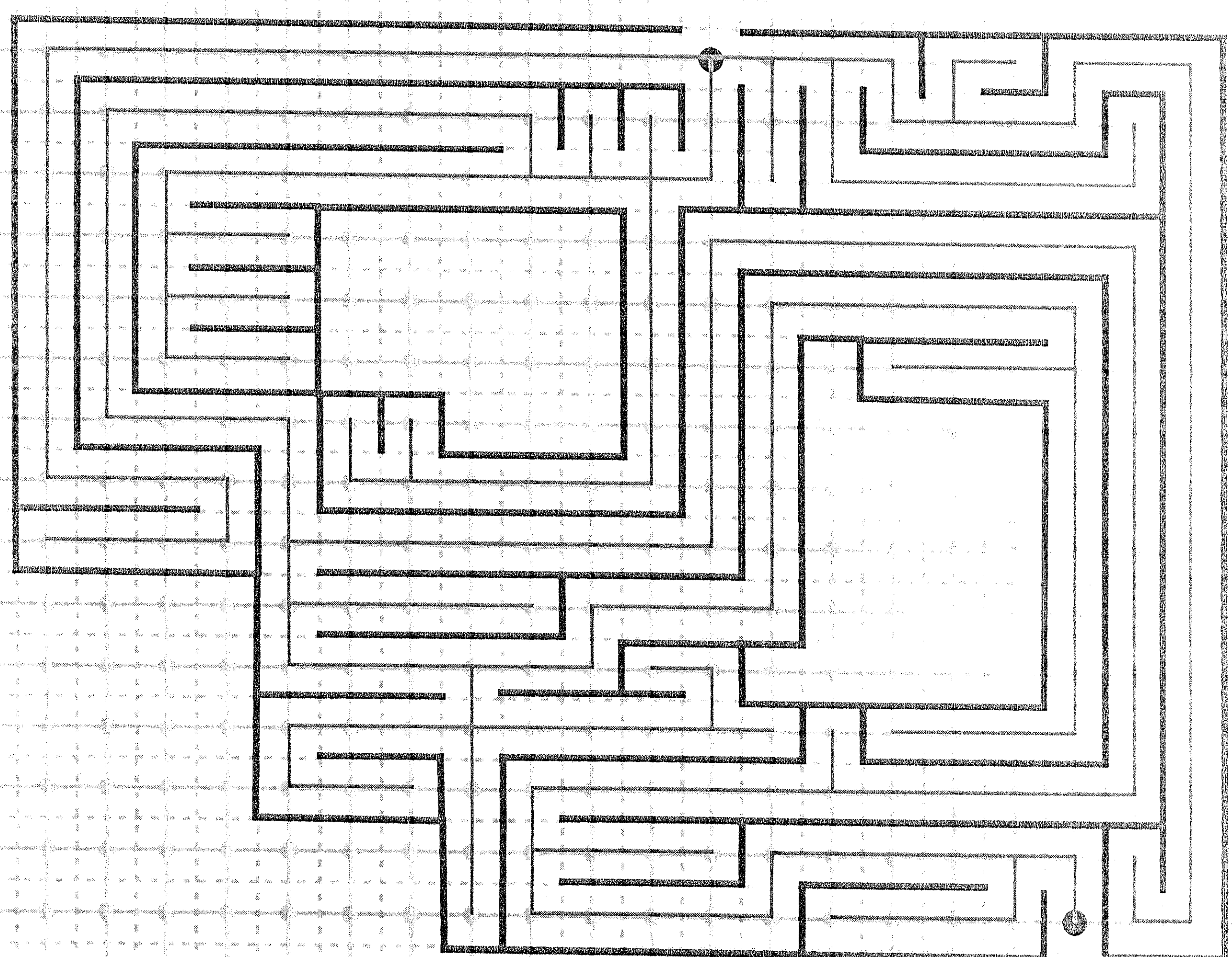


Step 2. Using a pencil, lightly draw a spanning tree containing all the vertices enclosed by the surrounding walls. (You will want to erase this graph after the maze is built around it, so do it lightly.)

A suggestion: start by drawing a path connecting the dot near the entry door to the dot near the exit door. This will end up as the solution path to the maze.



Step 3. Using a pen, draw all the remaining walls around the spanning tree, following along the dashed lines. Be careful not to cross any of the edges of the tree.



Step 4. Now erase the spanning tree to obtain your maze.

If you have tracing paper, you can copy the maze without the auxiliary grid.