(c) Copyright Kitty Meeks.  See www.dcs.gla.ac.uk/~kitty/GTotF for usage conditions.

# Vet vs Mosquito

Sometimes a disease can spread between animals even if they aren't close enough to pass it on directly – it might be spread by some other creature such as a mosquito, which files around biting various farm animals and spreading diseases around as it goes.

Today we're dealing with one particularly evil mosquito, who specifically wants to travel round all of our friends' farms to make all of the animals ill.  But luckily our friendly vet is going to help us out again: she's travelling round the farms at the same time with a vaccine to protect the animals.  The question is, will she be able to visit all of the farms before the mosquito has managed to make everyone ill?

Although the mosquito is very mean, it's not very clever: it can't plan its route ahead, so just visits one farm at a time and then decides where to go next.  The vet, on the other hand, has a map and can plan ahead to work out the best route to reach all the farms in the shortest possible time.

You need five volunteers to take the role of the farmers; the rest of the group should be divided into "mosquitos" and "vets".  Mosquitos and vets can either work as individuals or be split into small teams of up to four people.  You will need a printout of the file "Vet vs Mosquito – Farmers – Scenario X" for each scenario, a printout of "Vet vs Mosquito – Record Sheet" for each mosquito team and scenario (note this is the same for each scenario), and a printout of "Vet vs Mosquito – Vets – Scenario X" for each scenario and team of vets.

There are four scenarios with different travel times between the farms; you could repeat each of these with members of the group taking different roles.  Note that if you do repeat the scenario you will need an extra set of printouts for the vets and mosquitos (the farmer printouts can be reused).

Each of the farmers needs one of the sheets from the file "Vet vs Mosquito – Farmers – Scenario X"; the farmers should then spread themselves around the room and hold up their sheet of paper.  They should be close enough that they can easily read the name on everyone else's sheet, but far enough apart that they can't read the smaller writing underneath.  The mosquitos are going to move around the room, and they only learn the distances between each pair of farms when they visit one of the farms in question.  Make sure they don't get to read any of the information while the farmers are getting into position!

Each team of mosquitos is given a new copy of the record sheet ("Vet vs Mosquito – Mosquito Record Sheet") for each scenario.  They all start at Jim's farm; they will need to form a queue to start.  Their goal is to visit each farm and get back to Jim's farm.  Having read the information available at Jim's farm, the team decides where to go next; the farmer playing Jim will then cross off the number of boxes equal to the number of minutes needed to get to their next destination.  The team now walks to their next destination, and when they get there the relevant farmer writes their name in the next "route" box.  Now they can read the information at this farm and choose their destination; again, once they have chosen, the farmer crosses off the number of boxes equal to the number of minutes to get there.  The mosquito team keeps walking around following this procedure until they get back to Jim, having visited all the other farmers; Jim now writes his name in the last box.

Meanwhile, the vets are also trying to find a good route to visit all the farms, starting and finishing at Jim's, but they have access to all the information about distances at the start.  Give each team of vets a printout of "Vet vs Mosquito - Vets - Scenario X" for the current scenario; they then have to try to find the best route using pencil and paper while the mosquitos walk around the room.

(c) Copyright Kitty Meeks. See www.dcs.gla.ac.uk/~kitty/GTotF for usage conditions.

Once all the mosquitos have made it back to Jim's farm, ask everybody (mosquitos and vets) to announce their route, and how long it took (all of the group should check that the other teams' claimed routes do visit every farm and get back to Jim's, and that the times have been added up correctly). Keep a note of the best time achieved by both mosquitos and vets, and move onto the next scenario.

## Questions for discussion

- What methods did mosquitos use to decide where to go next?
- What methods did vets use to plan their route?
- Who did better on which examples?

## Solutions

The shortest routes in each scenario are as follows:

| Scenario | Route | Time (min) |
|---|---|---|
| 1 | Jim – Laura – Karl – Otto – Martin – Jim (or reverse direction) | 20 |
| 2 | Jim – Otto – Laura – Martin – Karl – Jim (or reverse direction) | 13 |
| 3 | Jim – Otto – Karl – Martin – Laura – Jim (or reverse direction) | 26 |
| 4 | Jim – Martin – Karl – Laura – Otto – Jim (or reverse direction) | 37 |

## The maths behind *Vet vs Mosquito*

What you saw here is an example of a famous problem called the **Travelling Salesman Problem**. In the Travelling Salesman Problem, a travelling salesman needs to visit a collections of cities before returning back home, and wants to find the shortest route by which he can do this – just the same goal as our mosquitos and vets. This is a very natural problem that arises in all sorts of real-life situations: think about a delivery driver who has fifty parcels to deliver today, and wants to work out the quickest way to do this and get back home. You can also think of it as the same problem if you want, for example, to work out the cheapest way to visit all the locations – if you've ever spent time looking at prices for train tickets, you'll know this isn't necessarily the same as the fastest or shortest route!

There are some obvious methods you could try to solve this, and you might have done this during the activity. For example, you could always go to the nearest location to where you are at the moment. This is an example of what's called a **greedy strategy**: you make the choice that looks best in the short term, but there's a risk you might end up stuck with only very bad options later on. This strategy will work well on some examples, but in other cases the route you take might be very much longer than is necessary.

The reason the Traveling Salesman Problem is so famous is that nobody knows a good way of solving it that will work quickly in all possible situations. We could of course work out the correct answer by considering all the possible orders in which he could visit the different cities, but the problem is that the number of different orderings grows extremely fast as we increase the number of cities. The table on the next page shows just how fast this number of possibilities grows.

Even with just five locations to visit, as in the activity, you probably wouldn't want to check all 120 possibilities by hand – but this wouldn't be too bad if you had a computer to do the work for you. But once there are 60 destinations – not an unreasonable number of stops for a delivery driver in one day – the number of possibilities is larger than the number of atoms in the known universe, so even with the fastest computer in the world you're going to have a very long wait.

(c) Copyright Kitty Meeks. See www.dcs.gla.ac.uk/~kitty/GTotF for usage conditions.

| Number of cities | Number of different orderings (to 3 significant figures) | Number of cities | Number of different orderings (to 3 significant figures) |
|---|---|---|---|
| 1 | 1 | 9 | 363,000 |
| 2 | 2 | 10 | 3,630,000 |
| 3 | 6 | 15 | $1.31 \times 10^{12}$ |
| 4 | 24 | 20 | $2.43 \times 10^{18}$ |
| 5 | 120 | 30 | $2.65 \times 10^{32}$ |
| 6 | 720 | 40 | $8.16 \times 10^{47}$ |
| 7 | 5040 | 50 | $3.04 \times 10^{64}$ |
| 8 | 40300 | 60 | $8.32 \times 10^{81}$ |

The problem here is that the number of possibilities to consider, if there are *n* locations to visit, is *n!* ("*n* factorial"), which is equal to n x (n-1) x (n-2) x … x 3 x 2 x 1, and this increases very fast as n grows. Certainly we're going to need more time to solve the problem as the number of locations increases – apart from anything else, it will take more time just to read all the information – but we'd like it if the amount of time depended on a *polynomial* function of *n*, for example $n^2$ or $n^3$. If we had a method where we only had to consider $n^3$ possibilities then the number of options to consider when we have 60 locations would be 216,000 – still far too many to check by hand, but okay with a computer.

Unfortunately, although it's possible (with some clever maths) to do a bit better than checking all of the *n!* possibilities, nobody knows a way to solve this problem that takes only a polynomial amount of time. In fact, this is one example of a whole family of natural mathematical problems where we don't know a way to find the answer that significantly quicker than checking all the possibilities. The interesting thing about this family of problems is that, if we found a good method to solve one of them, we could actually turn it into a good method to solve *all* of them – but so far nobody has managed to find such a method, and many people think that it's not possible to find a fast method to solve any of these problems.

However, this is a topic that it might be worth learning more about: so many mathematicians have got stuck on this question that, in the year 2000, the Clay Mathematics Institute included this problem (called **P versus NP**) as one of the seven **Millennium Mathematics Problems**. This means there's a $1 million prize if you either find an algorithm which will solve every possible example of the travelling salesman problem in polynomial time, or else prove that there can't be any algorithm that does this.