A humorous web based activity leads on to looking at ways to develop an algorithm to determine the shortest path between two points.

**Preparation required:**
Finding Kevin Bacon exercise sheet for each pupil or small group

# Six Degrees Of Kevin Bacon

This session starts with a fun activity that can be used to introduce another 'classic' algorithm and ends with a recap of how different data structures all 'fit together'.

Twenty years ago, inspired by "six degrees of separation," (the theory that nobody is more than six relationships away from any other person in the world) a game was dreamed up by movie buff Brian Turtle, called Six Degrees Of Kevin Bacon. The game involves trying to connect any movie star to the actor Kevin Bacon, linking them by other actors who have appeared in the same film. The presentation shows an example using Alexandra Daddario, who played Annabeth in Percy Jackson films. It explains how we calculate her Bacon Number as 2.

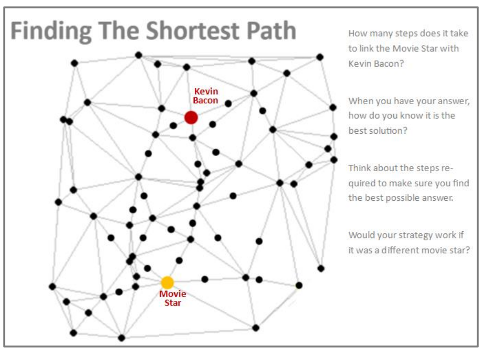
The internet movie database lists nearly 3 million actors and actresses, and about 2 million movies. A distribution of the statistics for 'Bacon numbers' is shown indicating that finding a movie star with a Bacon number of 4 or over is challenging! (Note the chart will be out of date as it is updated regularly). Such was the popularity of the game, Google have built Bacon number functionality into its search engine. Here's a fun class challenge! Pupils use Google to try to find a movie star with a Bacon number greater than 3. The star children search for must have appeared in a movie, not TV or music videos.

It's not just Google who calculate a movie star's Bacon Number. There is a dedicated website, called The Oracle of Bacon which does the same thing and more. The presentation uses the website to calculate the link between Daniel Radcliffe and Kevin Bacon, like shown previously in Google.

Although we have the same Bacon Number, the connections are different to those given in Google. Invite suggestions as to why this might be the case. The answer lies in understanding how a Bacon Number is calculated. So how does it do it?

All 3 million actors and actresses on the internet movie database are linked in a gigantic graph. Each node is a movie star. Each edge between two nodes is the film they both appeared in. An animation builds the links so Alexandra Daddario is linked to Logan Lerman (Percy Jackson) through the film Sea of Monsters. She is also linked to Douglas Smith, Katelyn Mager, Leven Rambin and every other actor/actress in that film. And of course, it is not just Alexandra who is linked to each star, but they are all linked to each other. The animation then links to stars who appeared in Noah… to give a sense of how complicated the graph will become.

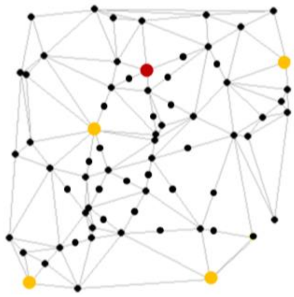
In reality, the graph is far more complicated than the animation, and would look more like the graphic in the presentation, with Kevin Bacon at the centre (the different colours represent people in the same film clustered together).




So a very large graph is the structure that holds all the data, but what about the algorithm to link a movie star? Let's try to work out an algorithm. As a whole class activity, with the diagram displayed as shown, imagine the red node is Kevin Bacon. The yellow node is another movie star entered on the website. We need to find the shortest path between the two.

Hand out the graph for students to try to find the smallest number of connections to link the two.

If students think they have found it, invite them to connect the path on the board. If they are struggling, announce you have found a link using eight steps – can they beat that? (They should, the shortest is 7 and the answer is given on the following slide).



Conclude by encouraging a class discussion. What lessons can we learn from this exercise? The best solution may not always be the most obvious. The shortest path starts by moving away from the destination for example. Ask how they got their answer. Did they develop a systematic approach? We are looking for them to articulate an algorithm that checks possible routes and is general enough to apply to any situation. Once students have articulated a possible approach, consider if it could be applied to the general situation of finding the shortest path for any Movie Star to Kevin Bacon (as shown by different yellow dots).

We can generalise the problem even further. The Oracle of Bacon allows you to calculate not just a star's Bacon number, but their degree of separation from any other movie star. So here we can check, for example the degree of separation of Alexandra Daddario from Daniel Radcliffe (Harry Potter). She has a Daniel Radcliffe Number of 2.

We can use a simpler graph to demonstrate the approach. The animation in the presentation walks through how a search builds up. From the starting node we identify every node connected to it and assign a value of 1. Then we take each node assigned a 1 in turn, identify every node connected to it, and label them with a value of 2 (if it has no value already assigned to it). A detailed explanation of the subsequent steps, and subtleties is in the slide notes. Check and rehearse carefully before using! It explains how Google and The Oracle of Bacon can get the same Bacon Number but with different links. It advances 'on click' so pause to ask pupils to predict which connections will be made if required.

# Finding The Shortest Path

Having found the destination, we can identify a shortest path. There may be others just as short, but no path can be shorter. An algorithm like this is known as a Breadth First Search because it searches every node one step away from the start, before it tries any further away.

Is Kevin Bacon at the centre of the Hollywood universe? To answer that involves calculating every star's Bacon number, then working out the average. The table on the slide shows the distribution. Kevin Bacon is, on average 3.009 degrees of separation from every other star. Challenge the pupils to find a star who has a lower average number. Who can find the best 'centre' of the movie universe?

Do read the explanation of how the site works. It gives an idea of the size of the graph structures that power applications such as these, as well as ways to link this work to other aspects of computing. Note in particular, the details about caching requests to speed up the ability to answer future similar queries. Calculating the shortest path to every movie star would take a long time!

Breadth first searches lie at the heart of shortest path algorithms. Such algorithms lie behind many common applications, such as Sat Nav's and route finding software. Encourage students to think about how the graph behind Google maps or AA route finder would be constructed. Every intersection of every tiny road is in there. The sheer size of these graphs makes route finding a demanding task. Shortest path algorithms have become much more efficient over the last 20 years. If they hadn't many route finding apps wouldn't be able to run on devices such as Smartphones which have far less memory than most PC's. Watch Simon Peyton-Jones (30 mins) talking about recent developments; [research.microsoft.com/apps/video/?id=146561](research.microsoft.com/apps/video/?id=146561)

The presentation ends with two exemplars from the Bebras competition 2015: [www.beaver-comp.org.uk](www.beaver-comp.org.uk). The first (p31) was aimed at KS3, a simple shortest path problem, which pupils can probably do in their head. The second (p59) is much more involved. Answers and explanations are contained in a separate handout. The complete booklet of questions is included in the resources, graded from youngest to oldest. Those on p25, 31, 34, 44, 47, 49, 50, 52, 55 and 59 are particularly pertinent to graphs and tree structures.