Queen Mary
University of London

The Australian Magician's Dream

Age group successfully used with:	11 – adult
Abilities assumed:	Nothing
Time:	20-30 minutes,
Size of group:	1 to 30 Larger groups also possible by using a web cam to project the table top onto a screen

Focus

What is an algorithm?
Search algorithms
Logical thinking and the correctness of algorithms
Computational Thinking

Syllabus Links

This activity can be used (for example)

- as a general introduction to computing topics such as what an algorithm is and how they can be compared from KS3 up.
- to introduce computational thinking problem solving from KS3 up,
- to teach specific syllabus topics such as:
  AQA A'level 3.1.1 Problem Solving: Linear Search
  KS3: understand several key algorithms (searching) that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem

Summary

Do a magic trick where your predict a card chosen that even the person choosing couldn't have known. Challenge the audience to work out how it is done, teach them how to do the trick and then use it to explain algorithms, searching, and logical reasoning.

Aims

This activity introduces a range of computing concepts in a memorable way. It introduces a magic trick and show how abstraction, logical reasoning and computational modelling can be used to check the algorithm works. Combined with the 'Punch card searching' activity it introduces a divide and conquer algorithm used by early computers.

Technical Terms

Search algorithms, abstraction, logical reasoning, computational modelling.

Materials

For the trick:

Pack of normal cards
Small table to do trick on.
A second 8 of hearts card (ideally large) to use as a prediction.
eg photocopy and enlarge an existing card.
Envelope for the second 8 of Hearts.
Mat or cloth to make it easier to handle cards on the table.
Special magic mats are available from magic shops.

To demonstrate a logical argument that it works (optional):

Cards with numbers 1 to 52 written on.
Number 16 should be on a different, brightly coloured card.
You can do it with a smaller number but at least 33.
A long coloured ribbon – 2-3m long
52 clothes pegs, one per card.

What to do

The Grab:

Announce you are going to teach the class a Computer Science magic trick. Because you are a teacher you will not only do the trick but teach them how to do it too. First though they must try and work out how it is done (if they think its not really magic). They also have to work out what it has to do with computing.

The Set-up:

For this trick, you need to do some set up before the audience arrive. You should have a normal pack of cards and then a second 8 of Hearts from another pack.

1) Take the pack of cards and find its 8 of Hearts and Ace of Hearts. Place the 8 of Hearts in the 16^{th} position in the pack. Place the Ace in the 32^{nd} position.
2) Place the second 8 of Hearts in an envelope. Put it on the floor, under the table where you will be doing the trick.

Put the mat on the table and check that it is easy to pick up cards. Place the pack face down on the table.

The activity:

Get a volunteer from the audience to the front. Spread the pack of cards face up on the table so that the full pack can be seen. Make sure that you can see the 8 and Ace of Hearts. Ask the volunteer to confirm that it is just a normal pack.

Say you need to roughly split the pack in two. As you do so hold your hands face up over the cards, apparently showing roughly the area where you need to split the pack. Secretly make sure that so that one hand is over the 8 of Hearts and the other over the 16 of Hearts. Ask the volunteer to POINT to a card somewhere near the middle and you will split the pack there. Don't say PICK a card as you don't want them to think that is the important card. You need the person to choose a card between the 16^{th} and 32^{nd} card, which is why you cover those cards with your hands giving an area to choose between them.

Now pick up all the cards to the right of that card – the bottom of the pack and put them aside. Confirm with the volunteer that it was their free choice of where you split the pack. Take the remaining cards, which will include the 8 of Hearts, and hold them face down.

Explain you are going to do a special deal, called the Down-Under deal. You know it because on nights before you do magic you always have strange dreams. In those dreams you see an Australian Magician who shows you tricks. Last night she showed you this special deal you are now going to do.

Say that this is how the Down-under deal works. Take the top card from the pack and place it on the table saying "Down". Then place the next card in a different pile face up and say "Under". Place the next card face down on the down pile, saying "Down". Keep going until you get to the end of the pack. Now explain you will always throw away the Down pile. Put it aside. Pick up the 'Under' pile which is left. It will contain the 8 of Hearts. Repeat the process with this 'under' pile, noting that the Down-under deal is the most boring deal in the world. Again discard the down pile and keep the Under pile. Continue doing this until there is only one card left face up on the table (it will be the 8 of Hearts).

Ask the volunteer to show everyone that card saying that it is the 8 of Hearts and ask them if they knew that was the one that would be chosen. Turn over the top few cards discarded and point out that if they had they split the pack anywhere else they would have ended up with a different card. Get them to confirm again that it was their free choice and it was a normal pack of cards.

Finally, announce that something amazing has happened because in your dream the Australian Magician also made a prediction. Ask the volunteer to look DOWN UNDER the table where they will find an envelope. Get them to take out the card in the envelope and show it to everyone. Amazingly (!) it is the same as the card the volunteer ultimately chose!

Thank the volunteer and ask everyone to give them a round of applause. Then challenge the audience to work out how it works (if it isn't really magic!)

Common suggestions of how it is done, include:

- "All the cards are the same."
  - Note that you might be able to make a trrick work like that but you didn't - show the full pack and point out the volunteer checked it was a normal pack.
- "You made them pick the right card."
  - A clever magician might be able to do that – Derren Brown perhaps – but you aren't that clever!
- "It must have been the last card."
  - This is on the right track, but point out that if it had been that card you would have discarded it straight away when the pack was cut.
- "It must have been the first card."
  - Again getting warmer, but that is the first card discarded in the down pile – show them!
- "It must have been the second card"
  - Getting the idea now but that card is discarded first on the second round of the deal. Agree it must have been in some fixed position... but where? You can let them keep trying to work it out, nudging them in the right direction or just congratulate them on getting close and then explain.

Point out that in coming up with ideas and ruling them out the students are doing some simple computational thinking, trying to come up with a method that would work. Where was the 8 of Hearts? It was in the 16th position from the top.

The explanation:
Magician's call this kind of trick a 'self-working' trick. If you follow the steps in the right order then the magical effect is guaranteed to happen. In this case the final selected card will be the 8 of Hearts. That is exactly what a computer scientist would call an algorithm. A simple version of the algorithm for this trick could be written out as:

1. Place the chosen card in position 16
2. Discard roughly the bottom half
3. Repeat 4 times:
    • Discard the first and then every second card thereafter
4. Reveal the card left is the one predicted.

Algorithms are just a series of steps to follow that guarantee to achieve some desired effect. In computing we mainly write algorithms for computers to follow. Magicians write them for other budding magicians to follow. The skill in coming up with a new magic trick is essentially the same as the skill in developing a new algorithm. Computer programs are just algorithms and in fact the above algorithm has a lot in common with a program. Magicians do computational thinking too – they have to do **algorithmic thinking**!

In the above description of the algorithm, we've actually used another computational thinking trick – **abstraction**. Rather than give all the details of the trick we've given a simplified version of some steps without all the detail. For example the step:

"Discard the first and then every second card thereafter"

doesn't include the detail of how that is done in the trick by dealing the cards into a Down pile and an Under pile then discarding the down pile. Abstraction allows us to focus on the big picture of how the algorithm works without getting lost in the details. We can worry about the details later.

The trick works because, as long as there are no more than 31 cards, by repeatedly discarding every second card you are guaranteed to end with the 16th card. How can we be sure of that? We can do some more computational thinking – **logical reasoning** in the form of a diagram as follows:

Originally:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24…51 52
After the cut:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
After deal 1
2 4 6 8 10 12 14 16 18 20 22 24
After deal 2
4 8 12 16 20 24
After deal 3
8 16 24
After deal 4
16
The card left is always the 16th card.

Let's explain this in more detail. We first represent the cards by their position from the top of the pack.

1 2 3 4 5 … 49 50 51 52

When the volunteer points to a card and we discard part of the pack we are getting rid of those from the end of the pack, from some point between the 16th and 32nd. That means we end up with the pack looking something like (again just showing positions)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Now if we remove the first and every second card (the ones underlined) we are getting rid of those in the odd positions. We are left with:

2 4 6 8 10 12 14 16 18 20 22 24

If we now do the same again, removing every second card we end up with those from the following original positions:

4 8 12 16 20 24

After the next deal we are left with just the 8 and the 16

8 16 24

and finally just the 16th card.

16

The card left is always the 16th card. This only works if there are no more than 31 cards there at the start, as otherwise the 32nd card would be left too and then on the next deal the 16th card would go leaving the 32nd.

The above argument is using another form of **abstraction**. In the argument we hide the details of the actual cards to make it simpler to follow. It doesn't matter what is on the cards. We therefore ignore that detail and instead represent them by just their original position in the pack. That way, whatever the cards actually are, we prove that the 16th card is the one left at the end.

This proof can be demonstrated physically too. In advance, peg between 33 and 52 numbered cards onto a long piece of ribbon. The cards should be placed in order with number 1 at the left and the largest at the right end. The 16th card should be on a different coloured card so it stands out.

Get two volunteers to hold the ends of the ribbon. Now follow the card trick's algorithm. Remove cards from the ribbon as they are discarded in the algorithm. First remove all those beyond the chosen point (which will be between positions 16 and 31). Explain what you are doing, and how this is what happens at the start when the person points to a card and that half of the pack is removed. Note that the 16th card is still in the same position from the top of the deck. Next remove the odd position cards starting at position 1 and saying "Down", "Under" as you alternatively remove and leave cards. Point out that the 16th card is left untouched as you go past it. Repeat this as in the trick removing every second card in repeated passes until only the 16th card is left.

This illustrates another aspect of computational thinking - **computational modelling**. The ribbon and cards are acting as a computational model of the pack in the trick. So was our series of diagrams. A **model** in this sense is just a simplified, usually mathematical, description of some phenomena. A **computational model** is one that you can run an algorithm on. Here we have created a physical computational model of the trick and 'run' it by having a person follow the steps. Usually computer scientists create computational models in software and running the computational model just involves running the program.

Computational models are a way to explore and understand an actual system. Here the system is a magic system: the deck of cards during the trick. A similar idea can be used to model all sorts of systems: biological systems (like the way heart or cancer cells work to better understand them) or financial systems (to explore the effects of traders behaving in particular ways for example). Once you have a computational model you can explore it by **simulating** it – running the algorithm as we did with the ribbon, or even proving properties about it.

This trick is not just a good illustration of the idea that self-working tricks are a kind of algorithm. A parallel (but otherwise identical) version of the algorithm used in the trick is the basis of the way early computers searched through data stored on punch cards. In the magic trick we searched for the 16^{th} card. In fact you can use variations of the down-under deal to end up with the card in any position you like, just by changing whether you discard the down or under piles at each step. This can be demonstrated using the linked "Punch card searching" activity (see end).

In summary, self-working tricks and computer programs are the same thing. Magicians who invent new tricks are doing the same computational thinking as those writing new programs: programmers really are wizards!

Variations and Extensions

Punch card searching

Combine this trick with the Punch card searching activity to give a direct link between the trick and computer search algorithms.

Prove it works

Set as a challenge for students in groups to come up with a convincing argument that the trick always works, before you go through a proof.

A student computational model

To illustrate the logical proof that the trick always works, use students standing in a line holding cards with their position written on to represent the pack of cards. Students sit down as their card is discarded. By using punch cards to do this, it could be combined with the Punch card search algorithm.

Find the 5^{th} card

Set as a challenge for students in groups to work out a version of the trick where you end up with the 5^{th} card every time rather than the 16th. HINT: You need to sometimes keep the DOWN pile and sometimes discard it.

Further Reading

The Magic of Computer Science

There are lots more magic tricks with computer science twists available from http://www.cs4fn.org/magic/ including several free magic books.

Links to other activities

The following activities are also available via teachinglondoncomputing.org

Punch card searching

Find out how binary numbers allow you to pull any card you want from a pile. This demonstrates a simple divide and conquer algorithm for searching, and in particular how early computers could quickly pull out any punch card from a pile. See how the algorithm is identical to the Australian Magician's dream magic trick.

20 Questions

Play 20 questions and show you can do divide and conquer problem solving. This introduces the idea of divide and conquer problem solving in the context of search algorithms. It also introduces the idea of efficiency analysis as a way of comparing algorithms.

Locked-in

Explore the design of an algorithm to allow someone who is totally paralysed to communicate.

This gives an introduction to computational thinking based problem solving, leading to an understanding of what an algorithm is and how algorithms can be compared on the basis of efficiency. It also illustrates how computational thinking is about more than just creating computer-based solutions. Computing is about solving problems for people.

Live demonstration of this activity

Teaching London Computing give live sessions for teachers demonstrating this and our other activities. See http://teachinglondoncomputing.org/ for details. Videos of some activities are also available or in preparation.

Department for Education

SUPPORTED BY MAYOR OF LONDON

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33	34
35	36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52