The Magic of Computer Science (Part 2)

Paul Curzon

Queen Mary University of London

teachinglondoncomputing.org/magicofcs/

Computer Science is about computation (not computers)

Magic = Computation

Fast Sorting and Searching

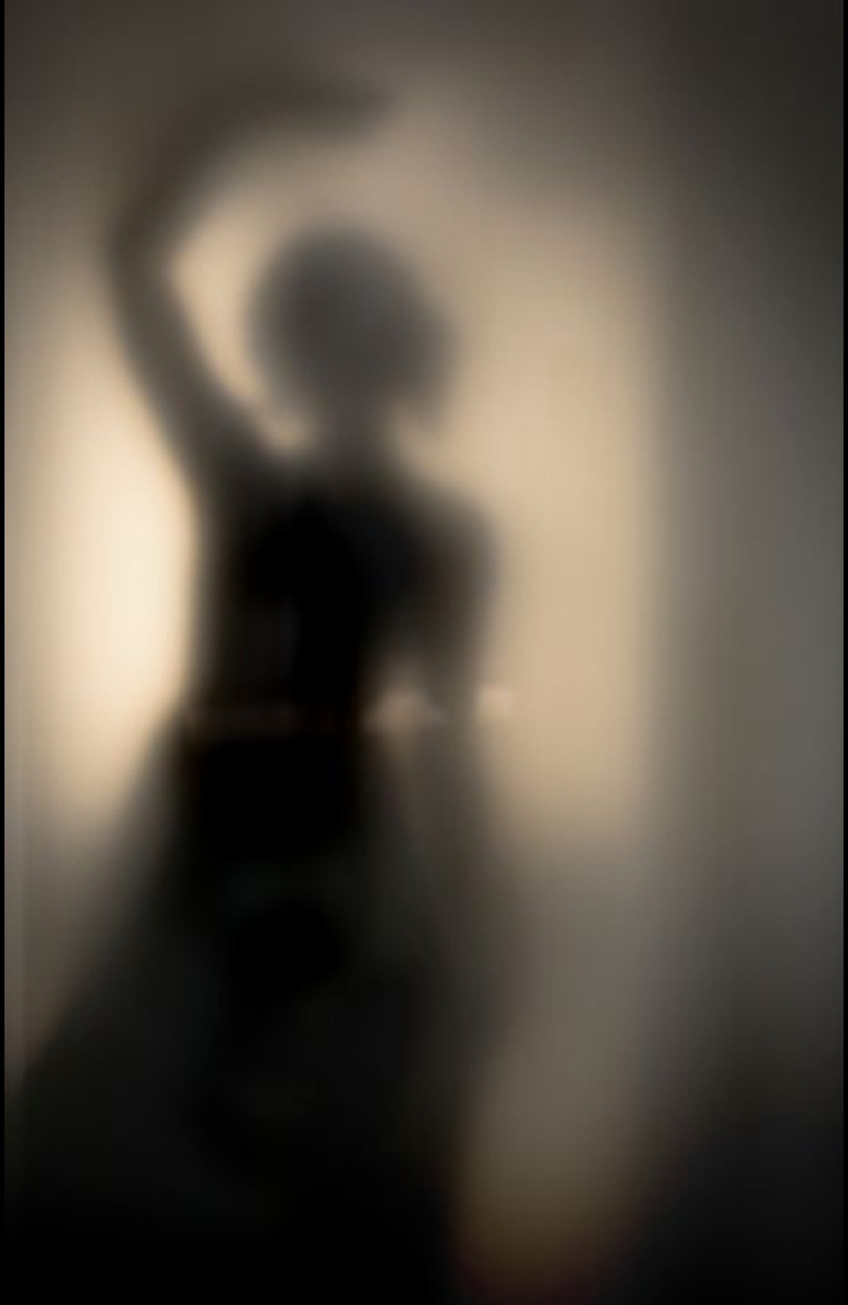
- We are going to explore how divide and conquer leads to fast algorithms
- Focus on search and sort algorithms
- Through Magic

On to the magic …
Please keep the secrets

- I’m going to teach you how to do the tricks
- Some are actually in the shows of professional magicians
- If you do perform them later for friends don’t break the magician’s code
- Keep the secrets!
- If you do know then don’t shout out let others puzzle it out first!

institute of CODING

The Australian magician’s dream



Queen Mary University of London

ALERT: A video message has arrived for you …

Ponder break

How on earth do they do that?

The computer science?

It is a self-working trick: an algorithm

```
1. Place the chosen card in position ???
2. Hide a prediction of that card (in an envelope)
3. Discard roughly the bottom half
4. Repeat 4 times:
       Discard the first and then
       every second card thereafter
5. Reveal the card is the one predicted.
```

Computers pull the same trick (algorithm) to search for data

Does it always work?

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,**16**,17,18,19,20,...

✹ ✹ ✹ ✹ ✹ ✹ ✹ ✹ ✹ ✹

2, 4, 6, 8,10,12,14,**16**,18,20,...

✹ ✹ ✹ ✹ ✹

4, 8,12,**16**,20,...

✹ ✹ ✹

8, **16**

✹

16

I set my force card 16 from the top

Try it yourself

The algorithm

1. Place the chosen card in position 16
2. Hide a prediction of that card (in an envelope)
3. Discard roughly the bottom half
4. Repeat 4 times:
   Discard the first and then
   every second card thereafter
5. Reveal the card is the one predicted.

It is a real search algorithm

- Early computers used punch cards as memory

- This trick is actually a real search algorithm
  - the algorithm was used by early computers to find data stored on punch cards

- Used to pull out a particular punch card based on numbers encoded in binary as holes and slots along the top of the card
  - A slot = 1
  - A hole = 0

We need some binary maths

16 in Binary is 10000

	16	8	4	2	1
x	1	0	0	0	0
=	16 +	0 +	0 +	0 +	0

5 in Binary is 00101

	16	8	4	2	1
x	0	0	1	0	1
=	0 +	0 +	4 +	0 +	1

0 means DISCARD the “down” pile
1 means KEEP the “down” pile

APPLY THIS RULE WITH PUNCH CARDS TO FIND ANY CARD

Searching for punch card 16

16 in Binary is 10000

1 :	0	DISCARD the “down” pile
2 :	0	DISCARD the “down” pile
4 :	0	DISCARD the “down” pile
8 :	0	DISCARD the “down” pile
16 :	1	KEEP the “down” pile

Searching for punch card 5

5 in Binary is 00101

1 :	1	KEEP the “down” pile
2 :	0	DISCARD the “down” pile
4 :	1	KEEP the “down” pile
8 :	0	DISCARD the “down” pile
16 :	0	DISCARD the “down” pile

Why is it fast?
Divide and Conquer

- Divide and Conquer algorithm
    - Halve the size of the problem with each step
    - Shake out half the cards
    - Based on 1 or 0 in next binary position
    - But left with the same problem (just fewer cards)
    - Stop when only 1 card left
- This is very very fast

Why is it fast? Divide and Conquer

- This is very very fast
    - Divide and conquer requires O(log n) operations
- log n is just the number of times we divide a number in half before getting down to 1
    - In this case 1 card left
    - So to search a million cards we would need only 20 operations
    - One for each bit in the binary

… leads to a sort algorithm

- On each pass
  - Instead of discarding the cards that drop
  - Place the discard cards at the back of the pile

- Leads to a lightning fast sort algorithm
- Divide and conquer gives O(n log n) sorting
  - eg Quicksort and Mergesort
  - see the separate video explanation of heap sort

How does that work?

Why is it fast? Parallelism

- It is even faster than that though because it uses parallelism too
  - based on the physicality of the cards
  - do all the comparisons to check if it is a 1 or a 0 in parallel
- would need a processor per card to get this extra speedup in a computer
- Makes it ridiculously fast
  - O(log n)
  - So (in theory)
    - can sort a million cards with 20 operations

The big picture?

Magic = secret method + presentation

Software = algorithm + interaction design

Both involve the same kind of
Computational Thinking

www.cs4fn.org/magic/

In summary

- Computation happens in the world (including in magic tricks)
- Divide and Conquer leads to fast algorithms
  - Binary representations are one basis for divide and conquer halving
- Parallelism also leads to fast algorithms
- Sorting and searching are good for exploring algorithms and efficiency

Dedicated to the
memory of
Peter McOwan

Thank you

Thanks to Richard Garriott for his part in the trick

Resources on magic and computer science at:

teachinglondoncomputing.org/magicofcs/