

# Computer Science... what's the big idea?

Tim Bell  
University of Canterbury  
Christchurch, New Zealand

Capital  
Scale 1:15,000,000  
Reference Projections  
standard parallels 38°N and 38°S



April 2004  
© 2004 by the National Geographic Society  
All rights reserved. No part of this publication may be reproduced without the written permission of the National Geographic Society.  
Bridges: representations of our world's architecture



Christchurch

Slovenia



Photo: Roy Sinclair  
<http://welcomeaboard.co.nz/blog/quaint-grocery-returns/>

# Example: Giving change

\$75 change?



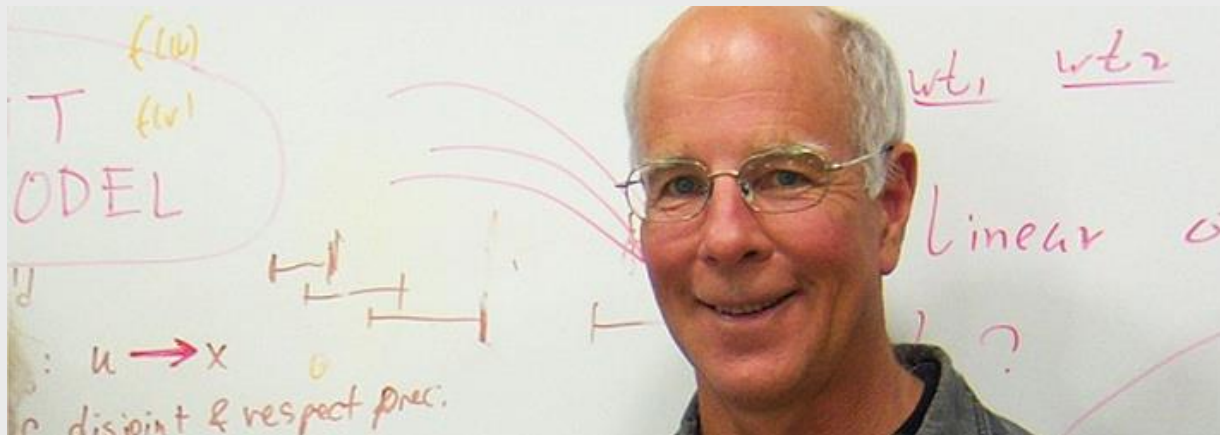
# Automated change



<http://www.stuff.co.nz/the-press/technology/the-box/the-ball/297404/Scan-pack-and-pay>

“Computer science is as much about programming as astronomy is about telescopes”

Mike Fellows



when  clicked

```
1 amount_owing = int(input("How much should I give out? "))
2
3 if 50 <= amount_owing:
4     print("here is $50")
5     amount_owing -= 50
6
7 if 20 <= amount_owing:
8     print("here is $20")
9     amount_owing -= 20
10
11 if 20 <= amount_owing:
12     print("here is $20")
13     amount_owing -= 20
14
15 if 10 <= amount_owing:
16     print("here is $10")
17     amount_owing -= 10
18
19 if 5 <= amount_owing:
20     print("here is $5")
```

say Here's \$20 for 2 secs

change amount owing by -20

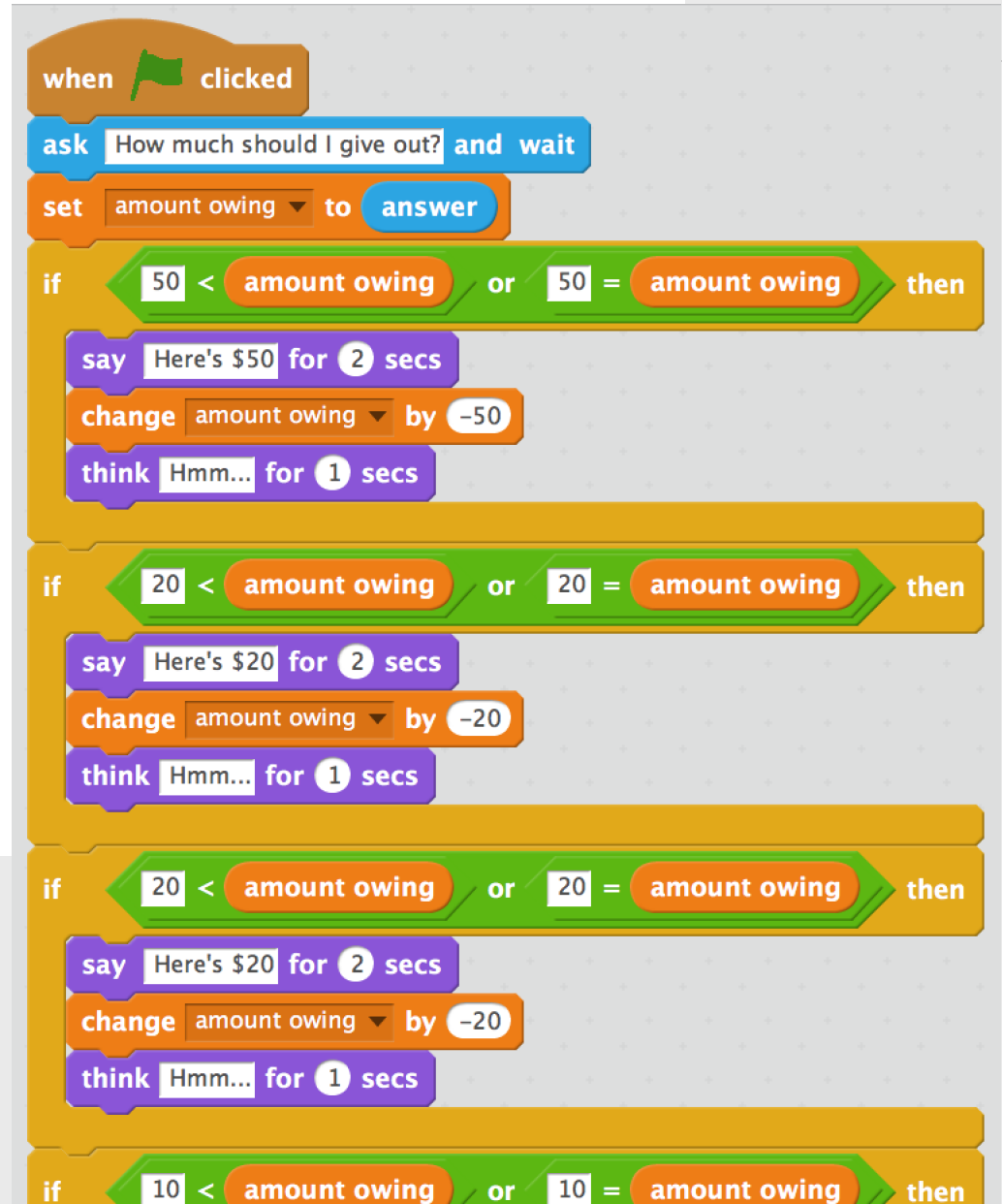
think Hmm... for 1 secs

if 10 < amount owing or 10 = amount owing then

```

1 amount_owing = int(input("How much should I give out? "))
2
3 if 50 <= amount_owing:
4     print("here is $50")
5     amount_owing -= 50
6
7 if 20 <= amount_owing:
8     print("here is $20")
9     amount_owing -= 20
10
11 if 20 <= amount_owing:
12     print("here is $20")
13     amount_owing -= 20
14
15 if 10 <= amount_owing:
16     print("here is $10")
17     amount_owing -= 10
18
19 if 5 <= amount_owing:
20     print("here is $5")

```



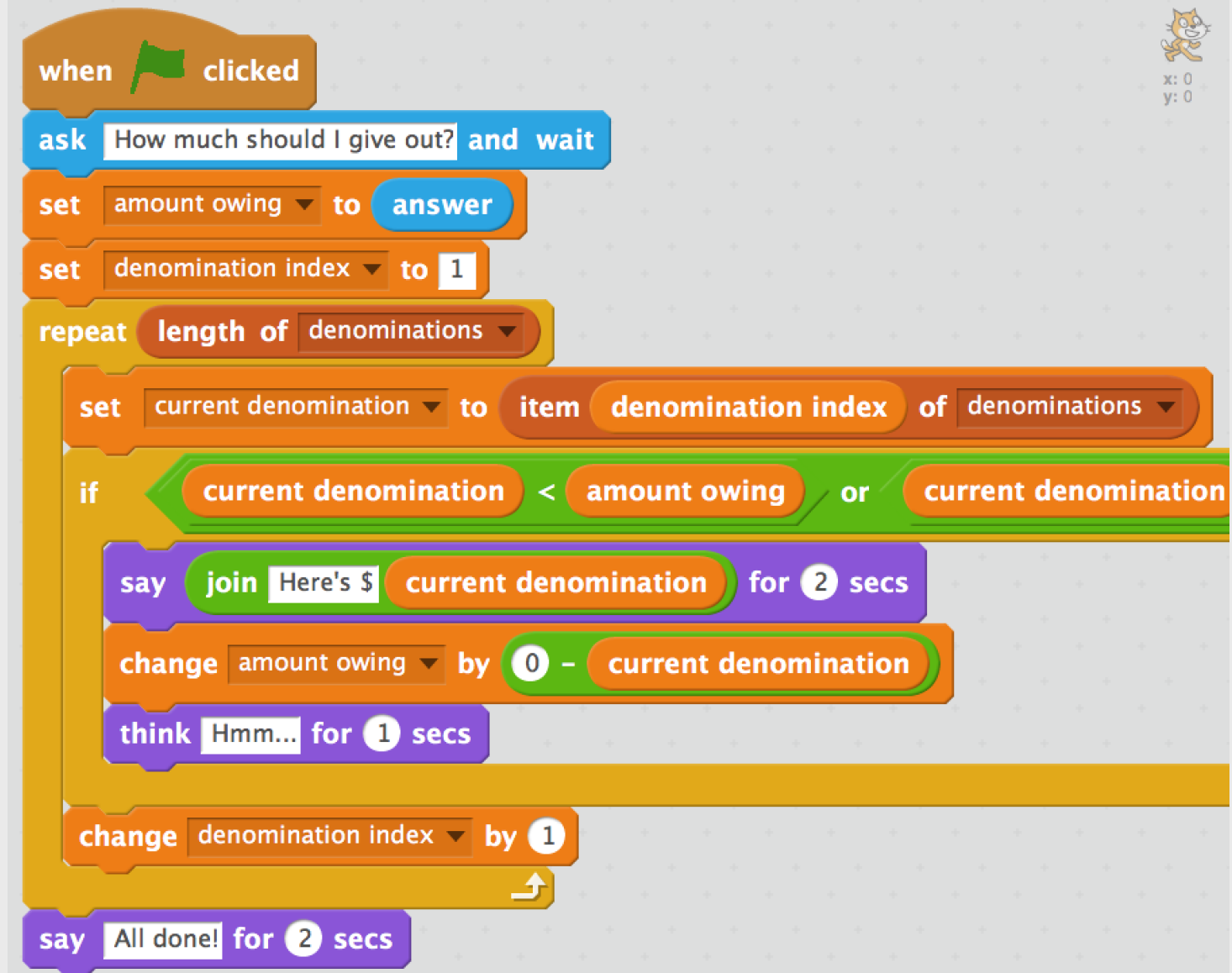
# Output

say Here's \$20 for 2 secs

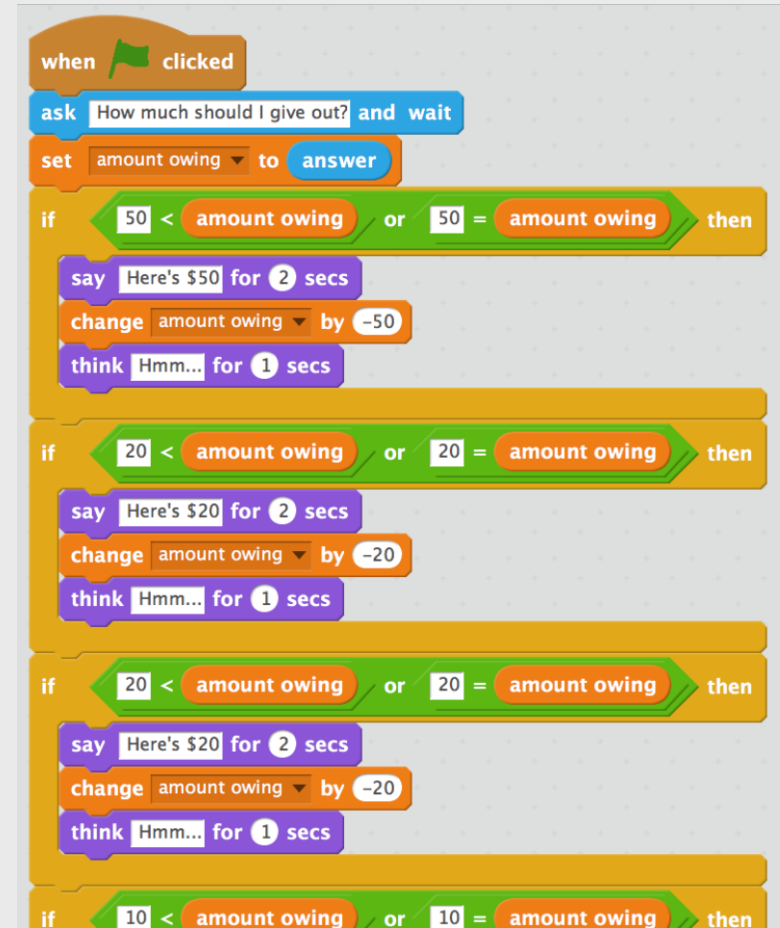
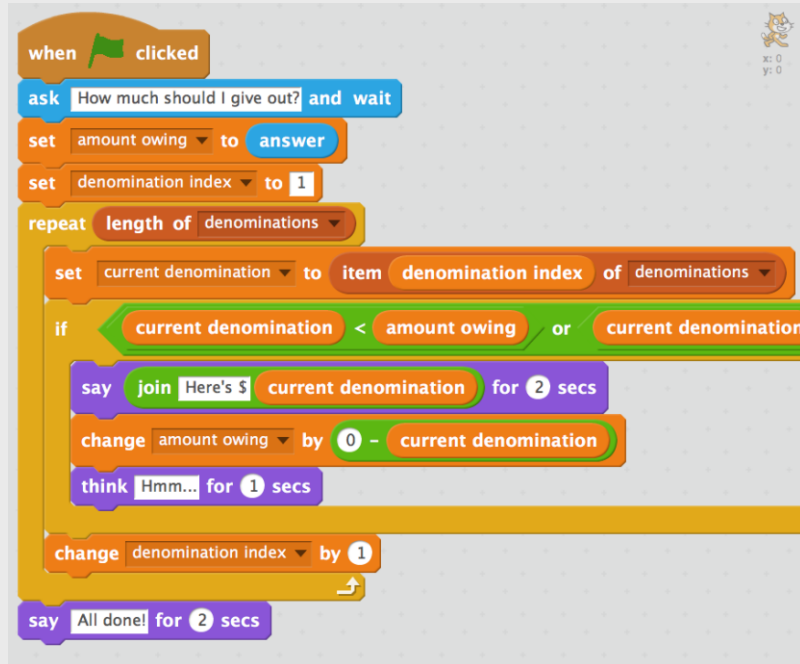
```
print("here is $20")
```



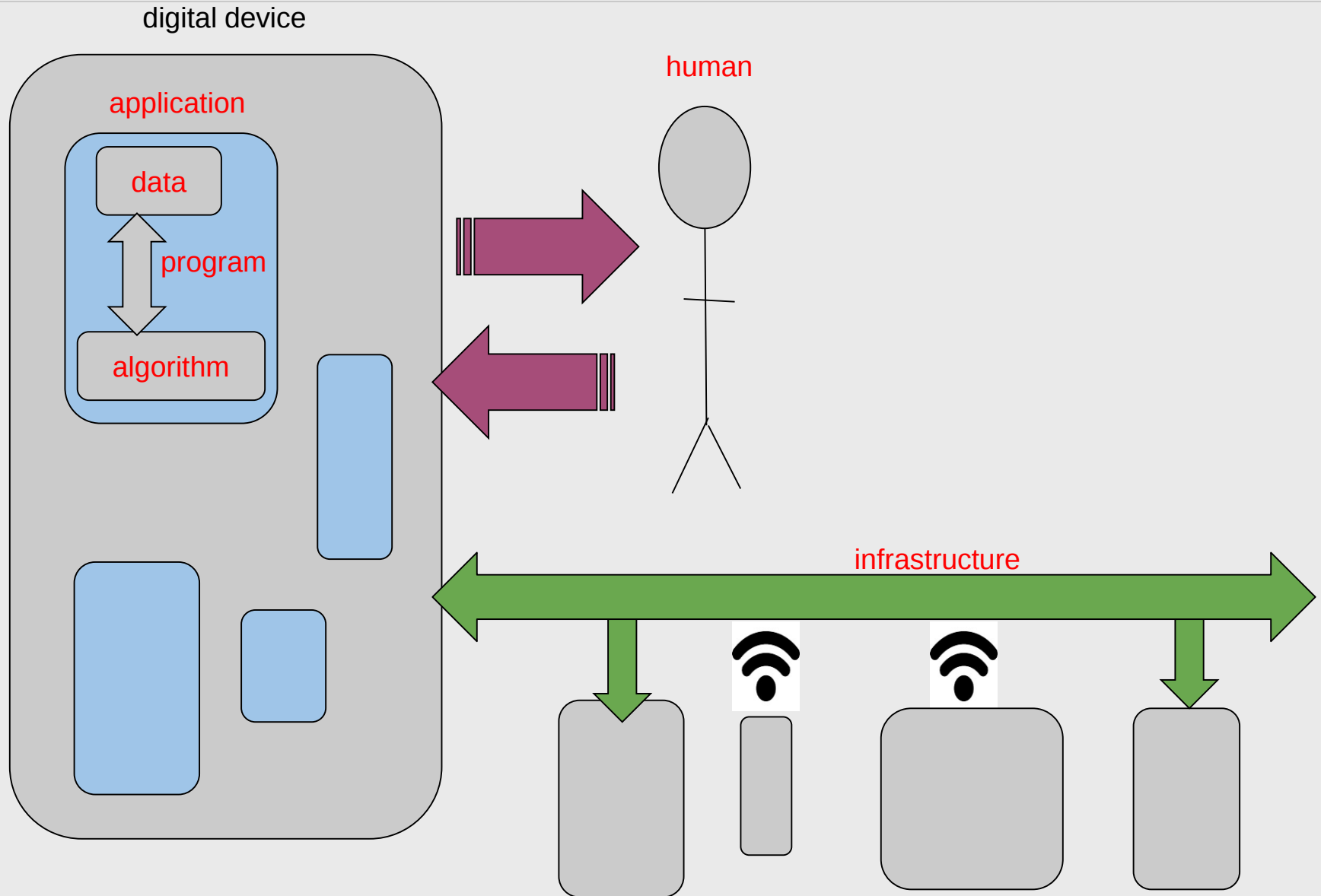
# Using iteration...



# There's no single correct answer...



# Structure of Digital Technologies



# Your digital day

---

- **Shopping**
- **Finance**
- **Transport**
- **Entertainment**
- **Privacy/security**
- **Fitness**

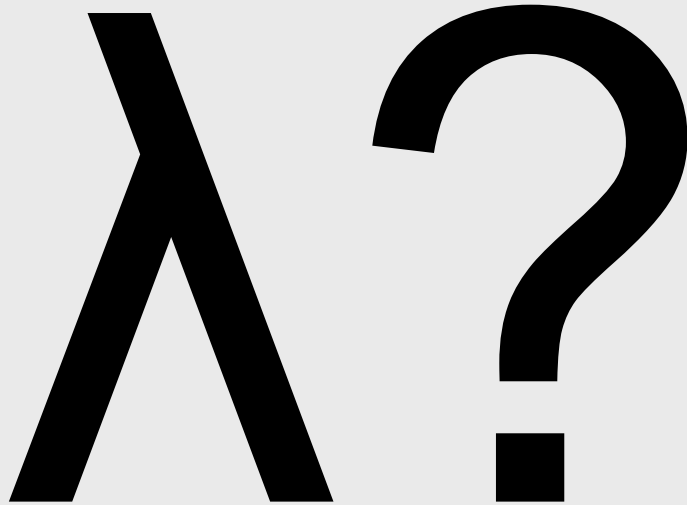
# Working with **Big Ideas** of Science Education

**Edited by Wynne Harlen**

with Derek Bell, Rosa Devés, Hubert Dyasi,  
Guillermo Fernández de la Garza, Pierre Léna,  
Robin Millar, Michael Reiss, Patricia Rowell and Wei Yu

the **key ideas** that students  
should **encounter** in their  
science education to  
enable them to **understand,**  
**enjoy and marvel** at the  
natural world

- to respond to students' perceptions of science as a **fragmented collection of facts and theories** of little relevance to them,
- help students to **explain things they find important**
- basis for **selection from the enormous range of possible curriculum content**
- inform the development of curriculum frameworks built on **progression towards big ideas**



the **key ideas** that students should **encounter** in their computing education to enable them to **understand, enjoy and marvel** at the digital world

<http://www.cosc.canterbury.ac.nz/research/RG/CSE/big-ideas/>

<https://goo.gl/mWXchU>

# The big ideas in CS/CT

1. **Data** is represented in digital form on a computer.
2. **Algorithms** interact with data to solve computational problems.
3. The **performance** of algorithms can be modelled and evaluated.
4. Some computational problems **cannot be solved** by algorithms.
5. **Programs** express algorithms and data in a form that can be implemented on a computer.
6. Computing systems are designed by **humans** to serve human needs.
7. Computing systems create **virtual** representations of natural and artificial phenomena.
8. **Protecting** data and system resources is critical in computing systems.
9. **Time dependent** operations in computing systems must be coordinated.
10. **Protocols** facilitate communication between computing systems.

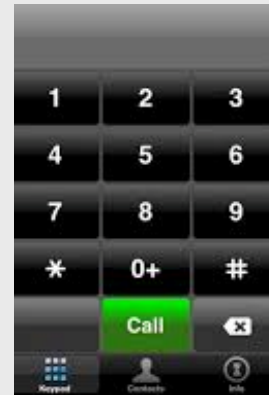
# **Data** is represented in digital form on a computer



0

1

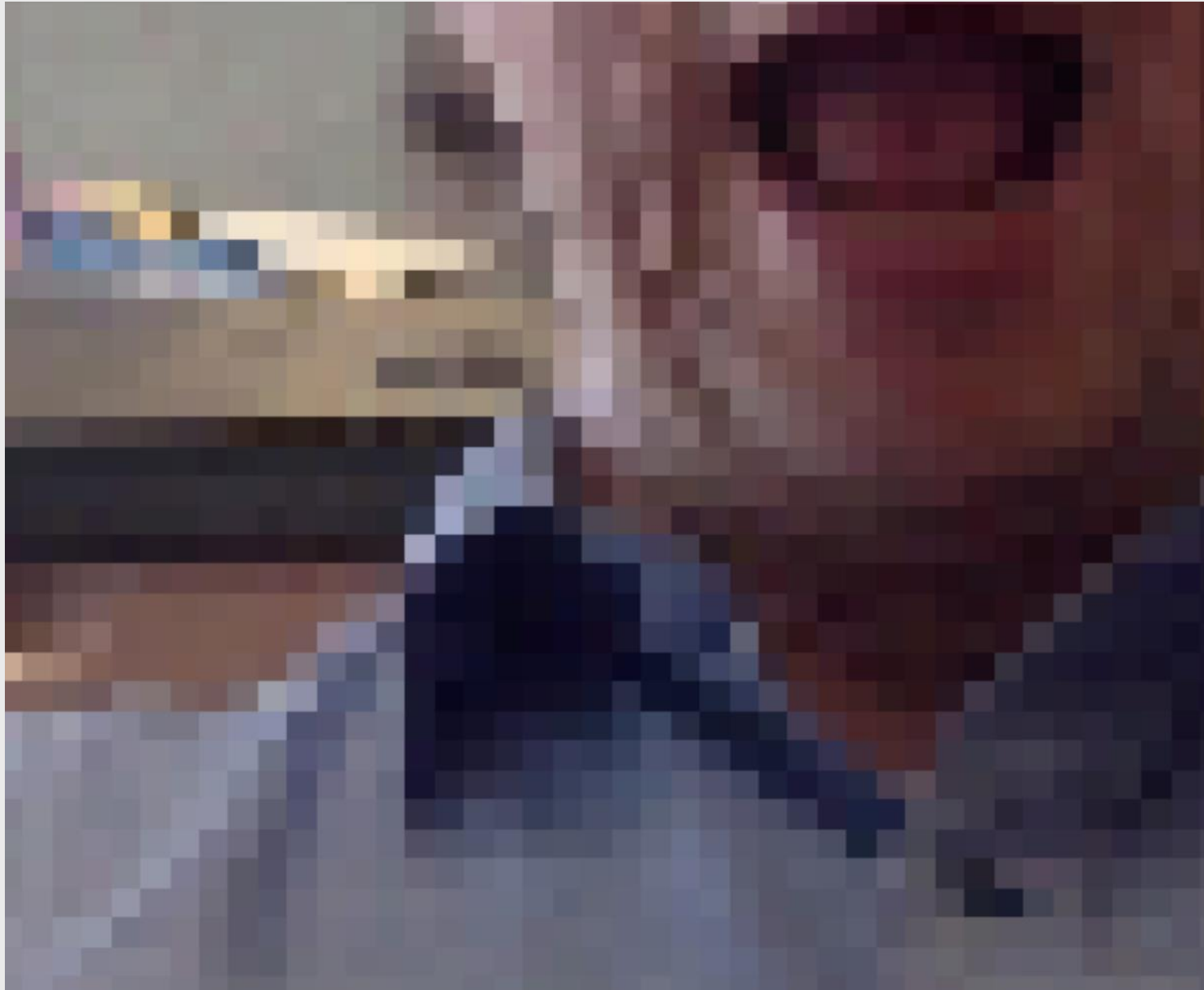
# It's all digital!



# Digital technologies?



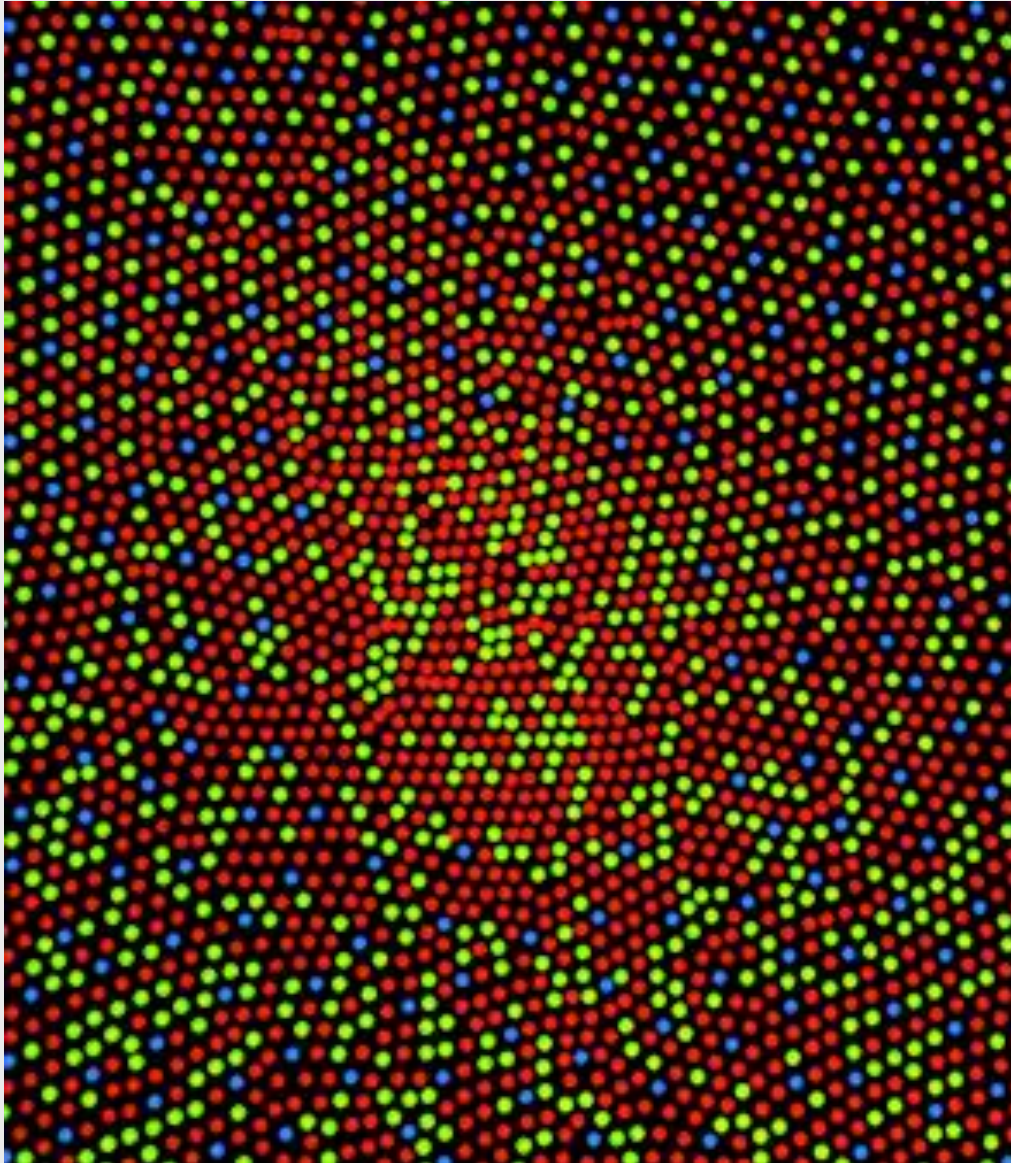
# Digital technologies?



# Digital technologies?

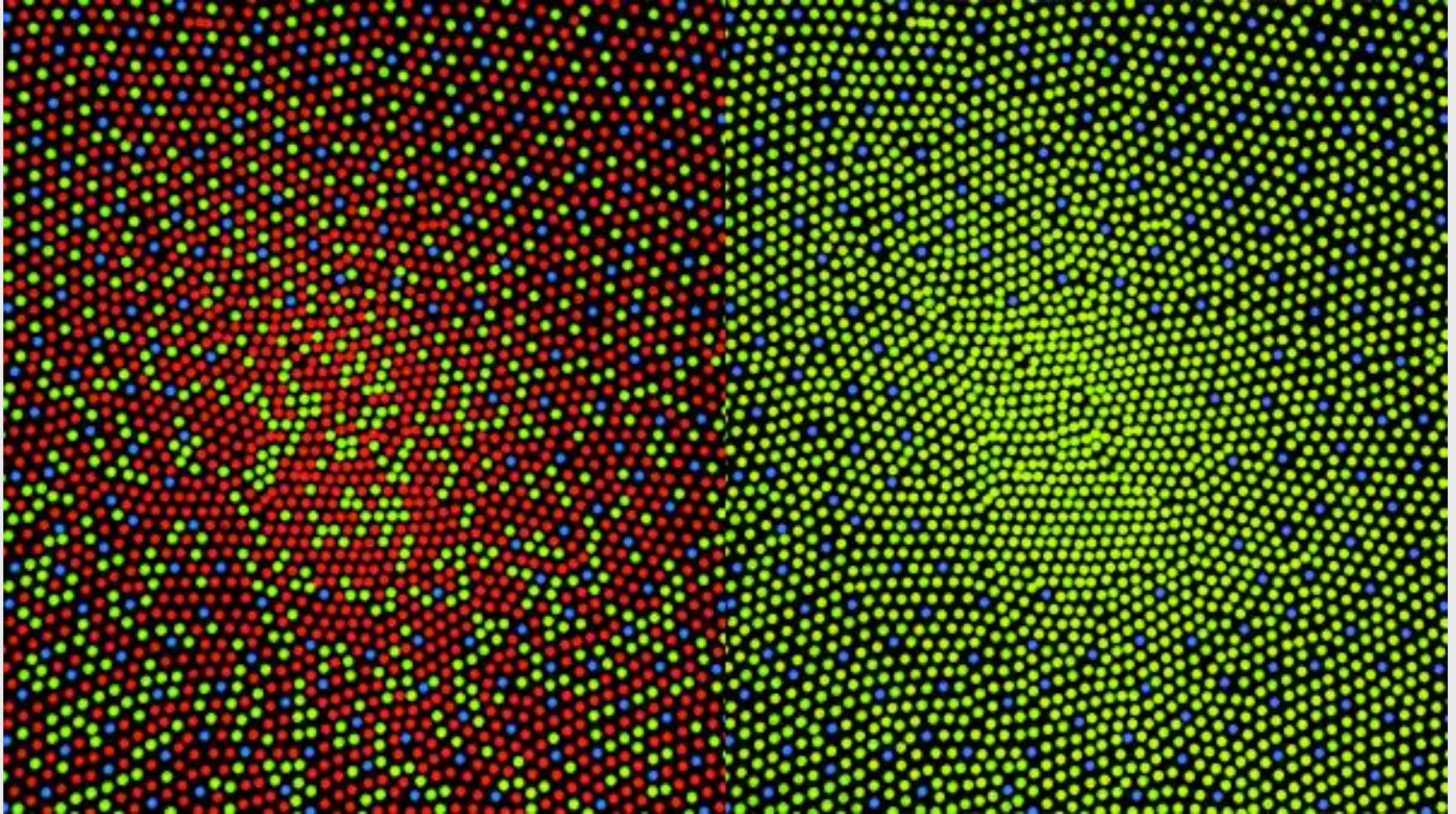
|       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R 86  | R 90  | R 85  | R 89  | R 89  | R 88  | R 92  | R 89  | R 97  | R 92  | R 82  |
| G 64  | G 68  | G 59  | G 60  | G 60  | G 60  | G 61  | G 57  | G 67  | G 63  | G 64  |
| B 66  | B 71  | B 60  | B 62  | B 58  | B 57  | B 61  | B 58  | B 65  | B 67  | B 70  |
| R 103 | R 106 | R 107 | R 104 | R 107 | R 110 | R 110 | R 111 | R 109 | R 113 | R 118 |
| G 79  | G 77  | G 78  | G 77  | G 78  | G 80  | G 78  | G 77  | G 80  | G 94  | G 106 |
| B 78  | B 72  | B 70  | B 69  | B 73  | B 72  | B 73  | B 68  | B 76  | B 101 | B 125 |
| R 124 | R 107 | R 108 | R 107 | R 111 | R 110 | R 110 | R 108 | R 130 | R 117 | R 73  |
| G 94  | G 76  | G 78  | G 80  | G 81  | G 80  | G 77  | G 78  | G 115 | G 114 | G 73  |
| B 91  | B 73  | B 72  | B 71  | B 71  | B 70  | B 71  | B 78  | B 131 | B 142 | B 98  |
| R 116 | R 109 | R 109 | R 111 | R 111 | R 111 | R 106 | R 120 | R 89  | R 67  | R 95  |
| G 83  | G 78  | G 79  | G 80  | G 81  | G 82  | G 78  | G 105 | G 93  | G 73  | G 97  |
| B 75  | B 72  | B 73  | B 70  | B 73  | B 76  | B 76  | B 116 | B 119 | B 97  | B 122 |
| R 100 | R 121 | R 118 | R 107 | R 105 | R 109 | R 145 | R 130 | R 74  | R 75  | R 103 |
| G 79  | G 106 | G 105 | G 86  | G 79  | G 97  | G 148 | G 133 | G 79  | G 82  | G 108 |
| B 81  | B 111 | B 105 | B 85  | B 80  | B 102 | B 160 | B 158 | B 109 | B 110 | B 129 |
| R 126 | R 120 | R 108 | R 110 | R 125 | R 126 | R 126 | R 68  | R 74  | R 98  | R 99  |
| G 124 | G 119 | G 107 | G 107 | G 123 | G 128 | G 130 | G 72  | G 74  | G 98  | G 101 |
| B 147 | B 137 | B 122 | B 123 | B 144 | B 156 | B 163 | B 101 | B 101 | B 121 | B 124 |
| R 115 | R 123 | R 124 | R 120 | R 125 | R 129 | R 97  | R 76  | R 81  | R 110 | R 101 |
| G 110 | G 119 | G 122 | G 120 | G 127 | G 134 | G 100 | G 76  | G 85  | G 108 | G 102 |
| B 131 | B 139 | B 139 | B 139 | B 149 | B 156 | B 121 | B 108 | B 113 | B 129 | B 126 |
| R 115 | R 117 | R 124 | R 123 | R 128 | R 113 | R 94  | R 68  | R 95  | R 107 | R 90  |
| G 114 | G 118 | G 123 | G 126 | G 133 | G 118 | G 95  | G 69  | G 94  | G 104 | G 92  |
| B 132 | B 136 | B 139 | B 141 | B 153 | B 140 | B 113 | B 99  | B 119 | B 125 | B 115 |

# Cone cells in the fovea



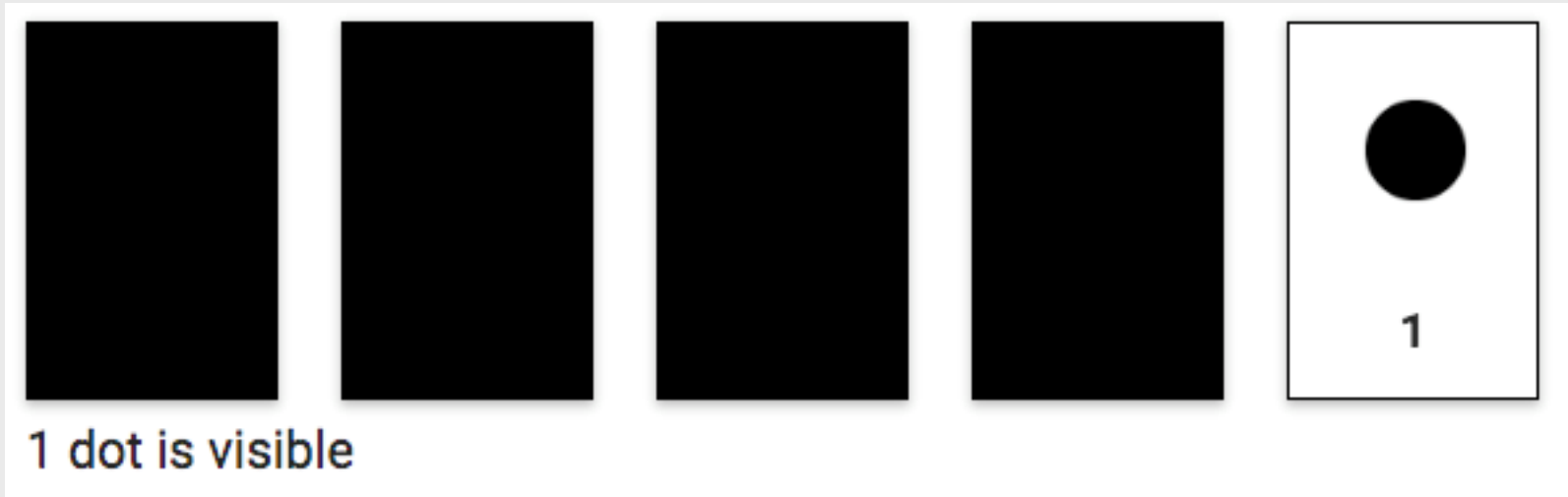
By Mark Fairchild, CC BY-SA 3.0,  
[https://commons.wikimedia.org/w/index.p  
hp?curid=32075732](https://commons.wikimedia.org/w/index.php?curid=32075732)

# Cone cells in the fovea

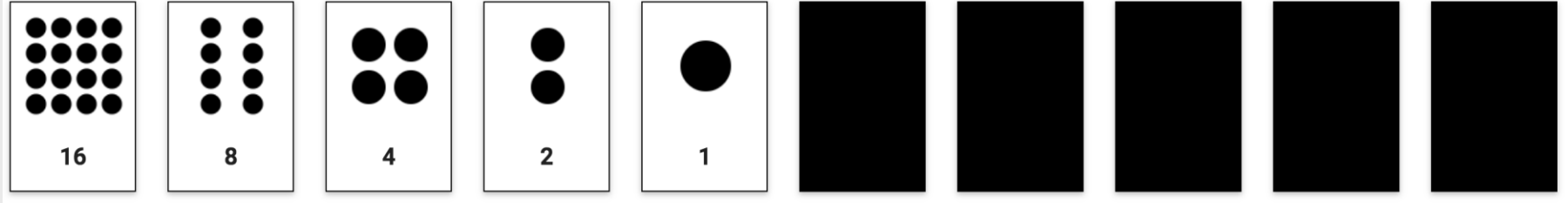


By Mark Fairchild, CC BY-SA 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=32075732>

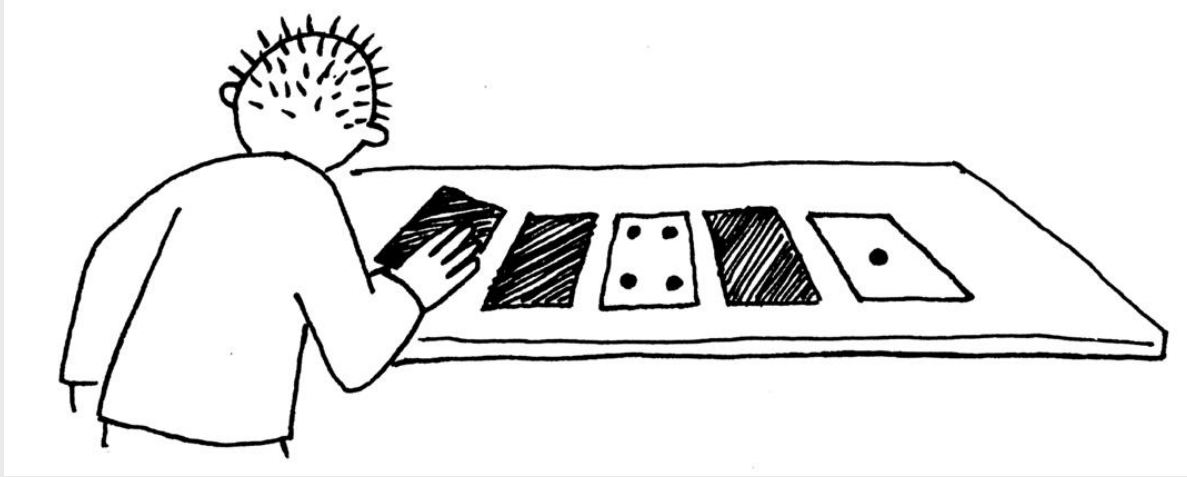
# Binary cards



# Binary cards



# Binary cards

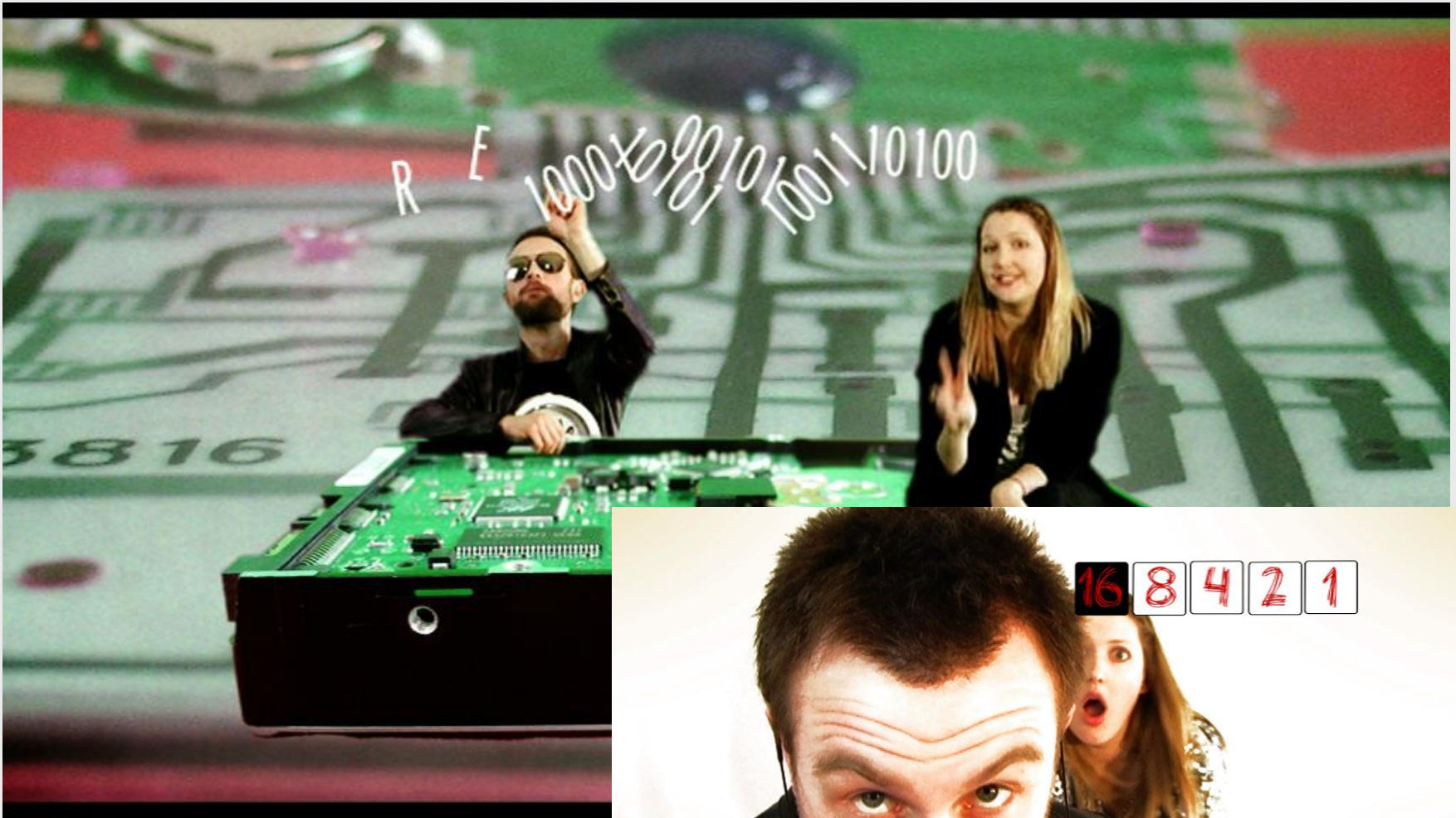


<http://csunplugged.org/binary-numbers/>

# Letters using digits

|               |                |                |
|---------------|----------------|----------------|
| 0 - 00000 -   | 9 - 01001 - I  | 18 - 10010 - R |
| 1 - 00001 - A | 10 - 01010 - J | 19 - 10011 - S |
| 2 - 00010 - B | 11 - 01011 - K | 20 - 10100 - T |
| 3 - 00011 - C | 12 - 01100 - L | 21 - 10101 - U |
| 4 - 00100 - D | 13 - 01101 - M | 22 - 10110 - V |
| 5 - 00101 - E | 14 - 01110 - N | 23 - 10111 - W |
| 6 - 00110 - F | 15 - 01111 - O | 24 - 11000 - X |
| 7 - 00111 - G | 16 - 10000 - P | 25 - 11001 - Y |
| 8 - 01000 - H | 17 - 10001 - Q | 26 - 11010 - Z |

# Unplugged: Modem



# Algorithms interact with data

to solve computational problems



# Algorithms interact with data

to solve computational problems



# Algorithms interact with data

to solve computational problems



# Algorithms interact with data

to solve computational problems

|       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R 121 | R 120 | R 116 | R 91  | R 96  | R 99  | R 97  | R 87  | R 89  | R 141 |
| G 31  | G 35  | G 51  | G 66  | G 73  | G 79  | G 81  | G 84  | G 107 | G 161 |
| B 40  | B 42  | B 46  | B 41  | B 41  | B 44  | B 43  | B 49  | B 80  | B 137 |
| R 120 | R 119 | R 128 | R 111 | R 89  | R 96  | R 80  | R 76  | R 150 | R 148 |
| G 32  | G 33  | G 38  | G 58  | G 72  | G 72  | G 83  | G 106 | G 164 | G 164 |
| B 42  | B 44  | B 45  | B 41  | B 38  | B 42  | B 51  | B 78  | B 144 | B 138 |
| R 120 | R 120 | R 130 | R 129 | R 94  | R 86  | R 78  | R 133 | R 143 | R 142 |
| G 35  | G 37  | G 41  | G 51  | G 62  | G 87  | G 104 | G 156 | G 155 | G 155 |
| B 44  | B 46  | B 51  | B 53  | B 42  | B 62  | B 83  | B 134 | B 135 | B 136 |
| R 128 | R 119 | R 108 | R 101 | R 81  | R 60  | R 128 | R 132 | R 136 | R 141 |
| G 56  | G 48  | G 45  | G 47  | G 83  | G 99  | G 153 | G 145 | G 150 | G 155 |
| B 67  | B 52  | B 37  | B 30  | B 60  | B 73  | B 129 | B 125 | B 130 | B 133 |
| R 168 | R 175 | R 170 | R 166 | R 63  | R 66  | R 93  | R 131 | R 132 | R 138 |
| G 136 | G 150 | G 151 | G 150 | G 68  | G 94  | G 121 | G 149 | G 146 | G 151 |
| B 87  | B 90  | B 81  | B 83  | B 23  | B 62  | B 97  | B 128 | B 121 | B 131 |
| R 171 | R 173 | R 174 | R 168 | R 151 | R 165 | R 78  | R 101 | R 127 | R 134 |
| G 162 | G 162 | G 164 | G 155 | G 133 | G 156 | G 88  | G 123 | G 145 | G 147 |
| B 95  | B 99  | B 100 | B 89  | B 60  | B 86  | B 46  | B 101 | B 125 | B 127 |
| R 173 | R 173 | R 173 | R 142 | R 164 | R 162 | R 165 | R 64  | R 102 | R 129 |
| G 165 | G 163 | G 160 | G 126 | G 149 | G 151 | G 152 | G 75  | G 127 | G 144 |
| B 107 | B 101 | B 95  | B 57  | B 80  | B 75  | B 83  | B 36  | B 105 | B 121 |

# Algorithms interact with data

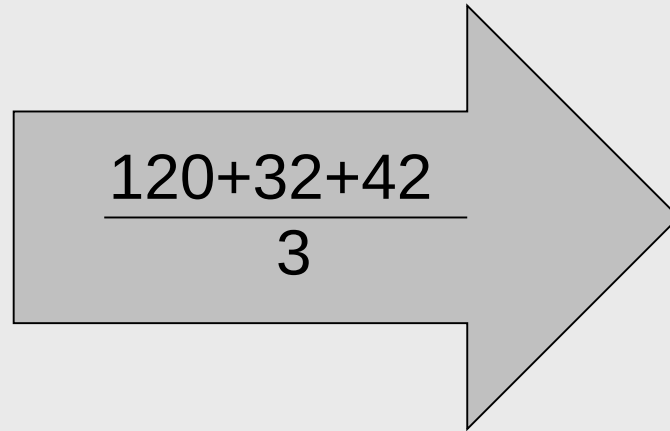
to solve computational problems

|       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R 63  | R 63  | R 70  | R 71  | R 76  | R 82  | R 82  | R 83  | R 101 | R 155 |
| G 63  | G 63  | G 70  | G 71  | G 76  | G 82  | G 82  | G 83  | G 101 | G 155 |
| B 63  | B 63  | B 70  | B 71  | B 76  | B 82  | B 82  | B 83  | B 101 | B 155 |
| R 63  | R 62  | R 68  | R 72  | R 74  | R 76  | R 81  | R 98  | R 160 | R 159 |
| G 63  | G 62  | G 68  | G 72  | G 74  | G 76  | G 81  | G 98  | G 160 | G 159 |
| B 63  | B 62  | B 68  | B 72  | B 74  | B 76  | B 81  | B 98  | B 160 | B 159 |
| R 63  | R 64  | R 71  | R 75  | R 69  | R 85  | R 97  | R 150 | R 150 | R 150 |
| G 63  | G 64  | G 71  | G 75  | G 69  | G 85  | G 97  | G 150 | G 150 | G 150 |
| B 63  | B 64  | B 71  | B 75  | B 69  | B 85  | B 97  | B 150 | B 150 | B 150 |
| R 78  | R 70  | R 64  | R 62  | R 80  | R 89  | R 146 | R 141 | R 146 | R 150 |
| G 78  | G 70  | G 64  | G 62  | G 80  | G 89  | G 146 | G 141 | G 146 | G 150 |
| B 78  | B 70  | B 64  | B 62  | B 80  | B 89  | B 146 | B 141 | B 146 | B 150 |
| R 141 | R 152 | R 151 | R 149 | R 65  | R 87  | R 113 | R 144 | R 141 | R 147 |
| G 141 | G 152 | G 151 | G 149 | G 65  | G 87  | G 113 | G 144 | G 141 | G 147 |
| B 141 | B 152 | B 151 | B 149 | B 65  | B 87  | B 113 | B 144 | B 141 | B 147 |
| R 160 | R 160 | R 162 | R 155 | R 133 | R 154 | R 83  | R 117 | R 140 | R 143 |
| G 160 | G 160 | G 162 | G 155 | G 133 | G 154 | G 83  | G 117 | G 140 | G 143 |
| B 160 | B 160 | B 162 | B 155 | B 133 | B 154 | B 83  | B 117 | B 140 | B 143 |
| R 162 | R 162 | R 159 | R 126 | R 149 | R 150 | R 151 | R 71  | R 121 | R 140 |
| G 162 | G 162 | G 159 | G 126 | G 149 | G 150 | G 151 | G 71  | G 121 | G 140 |
| B 162 | B 162 | B 159 | B 126 | B 149 | B 150 | B 151 | B 71  | B 121 | B 140 |

# Algorithms interact with data

to solve computational problems

|   |     |
|---|-----|
| R | 120 |
| G | 32  |
| B | 42  |



|   |    |
|---|----|
| R | 63 |
| G | 63 |
| B | 63 |



# Edge detection

|       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R 63  | R 63  | R 70  | R 71  | R 76  | R 82  | R 82  | R 83  | R 101 | R 155 |
| G 63  | G 63  | G 70  | G 71  | G 76  | G 82  | G 82  | G 83  | G 101 | G 155 |
| B 63  | B 63  | B 70  | B 71  | B 76  | B 82  | B 82  | B 83  | B 101 | B 155 |
| R 63  | R 62  | R 68  | R 72  | R 74  | R 76  | R 81  | R 98  | R 160 | R 159 |
| G 63  | G 62  | G 68  | G 72  | G 74  | G 76  | G 81  | G 98  | G 160 | G 159 |
| B 63  | B 62  | B 68  | B 72  | B 74  | B 76  | B 81  | B 98  | B 160 | B 159 |
| R 63  | R 64  | R 71  | R 75  | R 69  | R 85  | R 97  | R 150 | R 150 | R 150 |
| G 63  | G 64  | G 71  | G 75  | G 69  | G 85  | G 97  | G 150 | G 150 | G 150 |
| B 63  | B 64  | B 71  | B 75  | B 69  | B 85  | B 97  | B 150 | B 150 | B 150 |
| R 78  | R 70  | R 64  | R 62  | R 80  | R 89  | R 146 | R 141 | R 146 | R 150 |
| G 78  | G 70  | G 64  | G 62  | G 80  | G 89  | G 146 | G 141 | G 146 | G 150 |
| B 78  | B 70  | B 64  | B 62  | B 80  | B 89  | B 146 | B 141 | B 146 | B 150 |
| R 141 | R 152 | R 151 | R 149 | R 65  | R 87  | R 113 | R 144 | R 141 | R 147 |
| G 141 | G 152 | G 151 | G 149 | G 65  | G 87  | G 113 | G 144 | G 141 | G 147 |
| B 141 | B 152 | B 151 | B 149 | B 65  | B 87  | B 113 | B 144 | B 141 | B 147 |
| R 160 | R 160 | R 162 | R 155 | R 133 | R 154 | R 83  | R 117 | R 140 | R 143 |
| G 160 | G 160 | G 162 | G 155 | G 133 | G 154 | G 83  | G 117 | G 140 | G 143 |
| B 160 | B 160 | B 162 | B 155 | B 133 | B 154 | B 83  | B 117 | B 140 | B 143 |
| R 162 | R 162 | R 159 | R 126 | R 149 | R 150 | R 151 | R 71  | R 121 | R 140 |
| G 162 | G 162 | G 159 | G 126 | G 149 | G 150 | G 151 | G 71  | G 121 | G 140 |
| B 162 | B 162 | B 159 | B 126 | B 149 | B 150 | B 151 | B 71  | B 121 | B 140 |

# Canny Unplugged

Try it yourself! The grid below is filled with numbers that represent a grayscale image. See if you can detect edges the way a computer would do it. **If the values of two neighboring squares on the grid differ by more than 50, draw a thick line between them.**

|    |    |    |     |     |     |     |     |     |     |     |     |     |    |     |    |
|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|-----|----|
| 21 | 21 | 21 | 21  | 21  | 21  | 21  | 21  | 21  | 21  | 21  | 21  | 21  | 21 | 21  | 21 |
| 21 | 21 | 21 | 21  | 21  | 21  | 21  | 103 | 103 | 21  | 21  | 21  | 21  | 21 | 21  | 21 |
| 21 | 21 | 21 | 21  | 21  | 21  | 21  | 99  | 103 | 21  | 21  | 21  | 21  | 21 | 21  | 21 |
| 21 | 21 | 21 | 21  | 21  | 21  | 21  | 91  | 103 | 21  | 21  | 21  | 21  | 21 | 21  | 21 |
| 21 | 21 | 21 | 21  | 21  | 21  | 21  | 103 | 99  | 21  | 21  | 21  | 21  | 21 | 21  | 21 |
| 21 | 21 | 21 | 21  | 21  | 21  | 91  | 99  | 103 | 99  | 21  | 21  | 21  | 21 | 21  | 21 |
| 21 | 72 | 93 | 101 | 103 | 105 | 99  | 103 | 91  | 99  | 104 | 102 | 101 | 91 | 103 | 21 |
| 21 | 21 | 99 | 99  | 91  | 103 | 99  | 99  | 103 | 99  | 99  | 91  | 99  | 88 | 21  | 21 |
| 21 | 21 | 21 | 99  | 99  | 91  | 91  | 99  | 103 | 99  | 91  | 99  | 103 | 21 | 21  | 21 |
| 21 | 21 | 21 | 21  | 99  | 99  | 99  | 99  | 99  | 99  | 99  | 103 | 21  | 21 | 21  | 21 |
| 21 | 21 | 21 | 21  | 21  | 99  | 103 | 103 | 91  | 103 | 103 | 21  | 21  | 21 | 21  | 21 |
| 21 | 21 | 21 | 21  | 21  | 103 | 91  | 99  | 99  | 103 | 91  | 21  | 21  | 21 | 21  | 21 |
| 21 | 21 | 21 | 21  | 103 | 91  | 99  | 99  | 103 | 99  | 103 | 99  | 21  | 21 | 21  | 21 |
| 21 | 21 | 21 | 21  | 91  | 103 | 99  | 21  | 21  | 99  | 91  | 99  | 21  | 21 | 21  | 21 |
| 21 | 21 | 21 | 103 | 103 | 21  | 21  | 21  | 21  | 21  | 21  | 103 | 99  | 21 | 21  | 21 |
| 21 | 21 | 21 | 103 | 3   | 21  | 21  | 21  | 21  | 21  | 21  | 7   | 103 | 21 | 21  | 21 |

# Algorithms interact with data

to solve computational problems



Add 1



# Example: Giving change

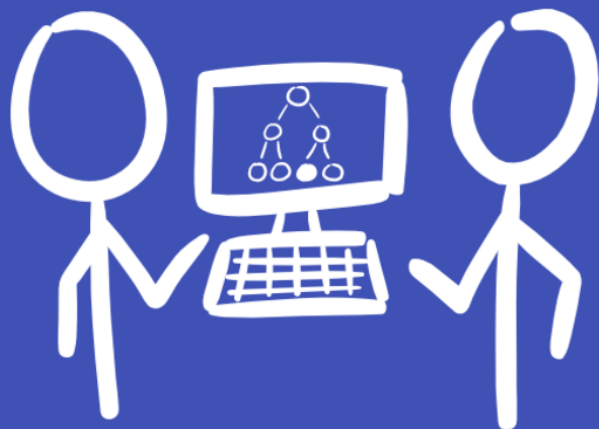
\$75 change?



# The **performance** of algorithms can be modelled and evaluated



<https://www.youtube.com/watch?v=iDVH3oCTc2c>



# Computer Science Field Guide

- Algorithms
- Programming Languages
- Human Computer Interaction
- Data Representation
- Coding - Introduction
- Coding - Compression
- Coding - Encryption
- Coding - Error Control
- Artificial Intelligence
- Complexity and tractability
- Computer Graphics
- Computer Vision
- Formal Languages
- Network Communication Protocols
- Software Engineering

# CSFG searching



Box 1



Box 2



Box 3



Box 4



Box 5

# Searching



# Searching Phone Book

- **512 pages**

256 pages,

128 pages,

64, 32, 16, 8, 4, 2, 1

**9 rips**

- **1024 pages**

512 pages,

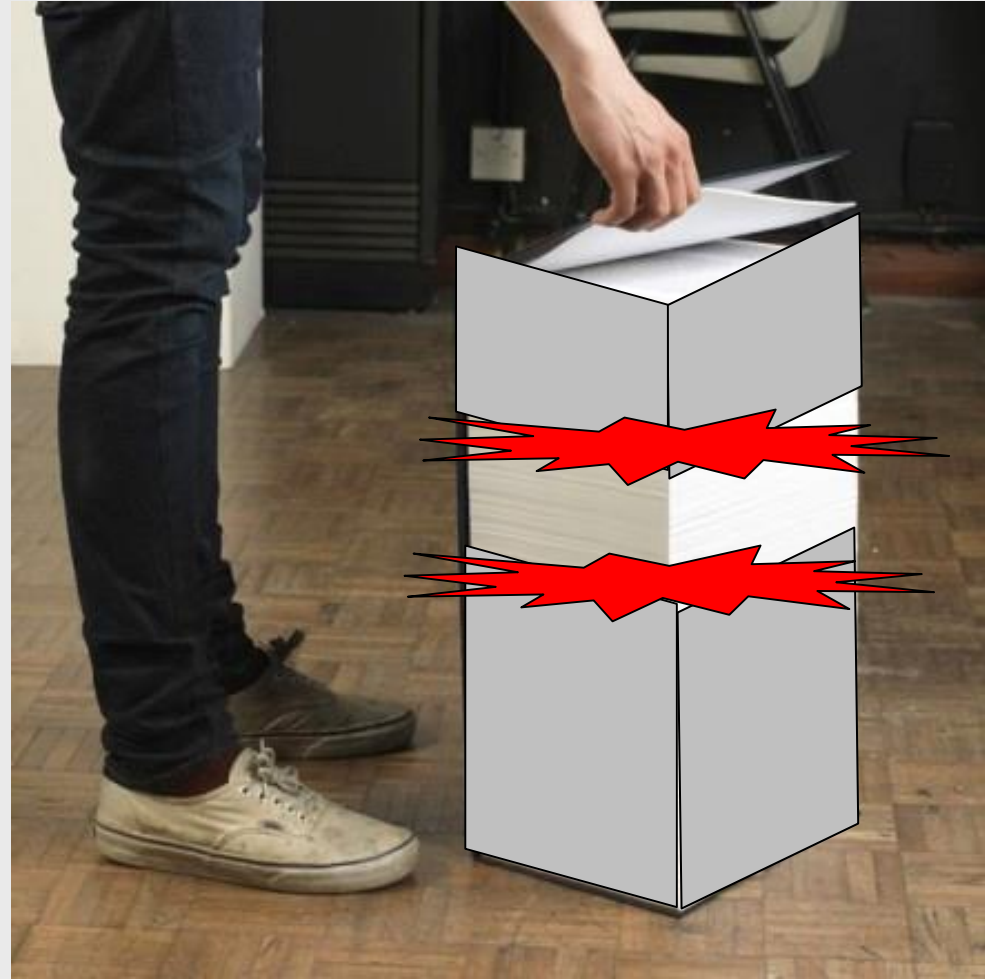
256 pages,

128, 64, 32, 16, 8, 4, 2, 1

**10 rips**

# 5000 pages

- Wikipedia featured articles
- 2500 pages,
- 1250 pages,
- 625 pages ...
- (13 rips)



Rob Matthews

<http://www.geek.com/geek-cetera/0-01-of-wikipedia-turned-into-5000-page-book-1336419/>

# 1,000,000 pages

- 500,000,  
250,000...  
... 1953,  
976  
(10 rips)



Pedia Press  
<http://hexus.net/business/news/internet/66305-wikipedia-encyclopaedia-become-million-page-book/>

- 20 rips total

# 1,000,000,000 pages

50 km long

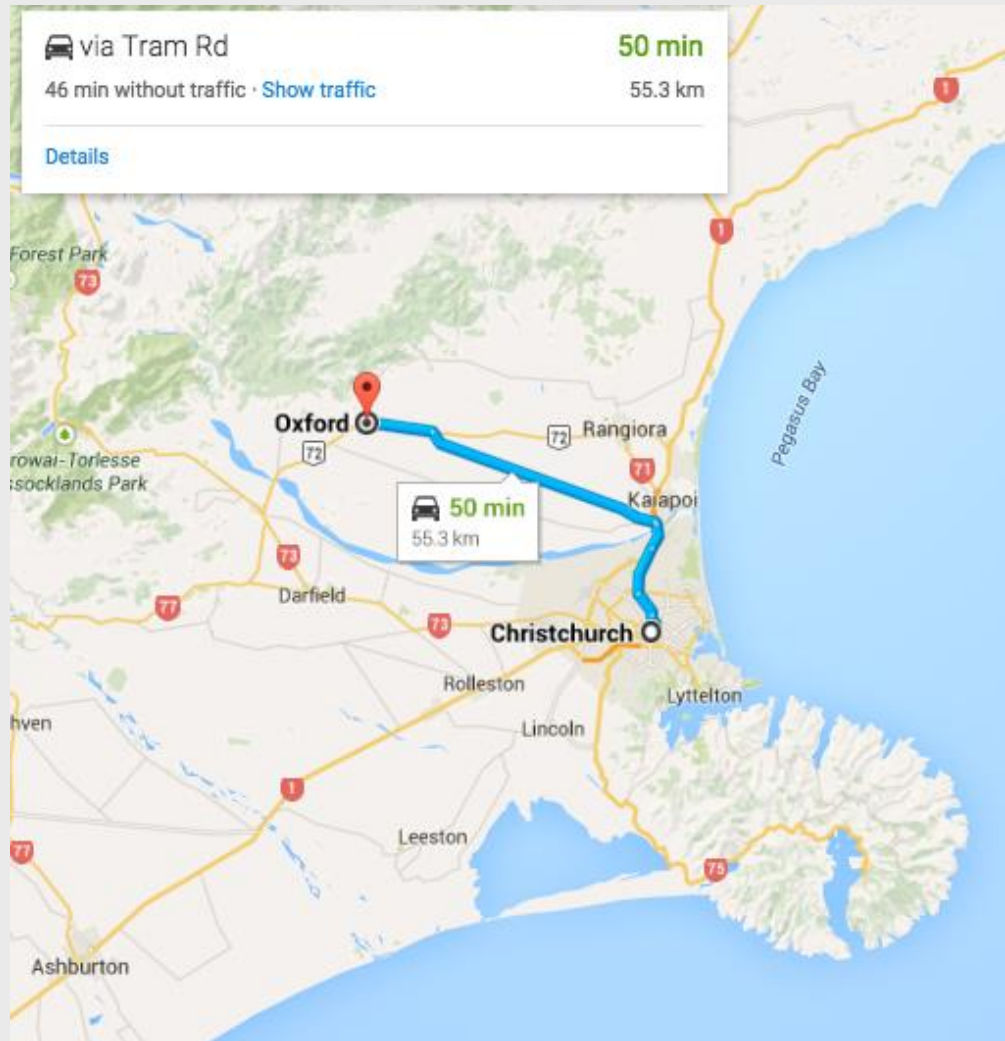
25km,

12.5km

6.25 km

Total?

30 rips

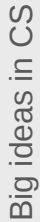


# Big ideas in CS

# Big ideas in CS

# Big ideas in CS

# Big ideas in CS



# Telogis routing

Customers Routes Debug Tools Testing Tools Help ☐ Show Territories

Week Starting Monday, 16 November 2009

Territories Balancing Routes Problem

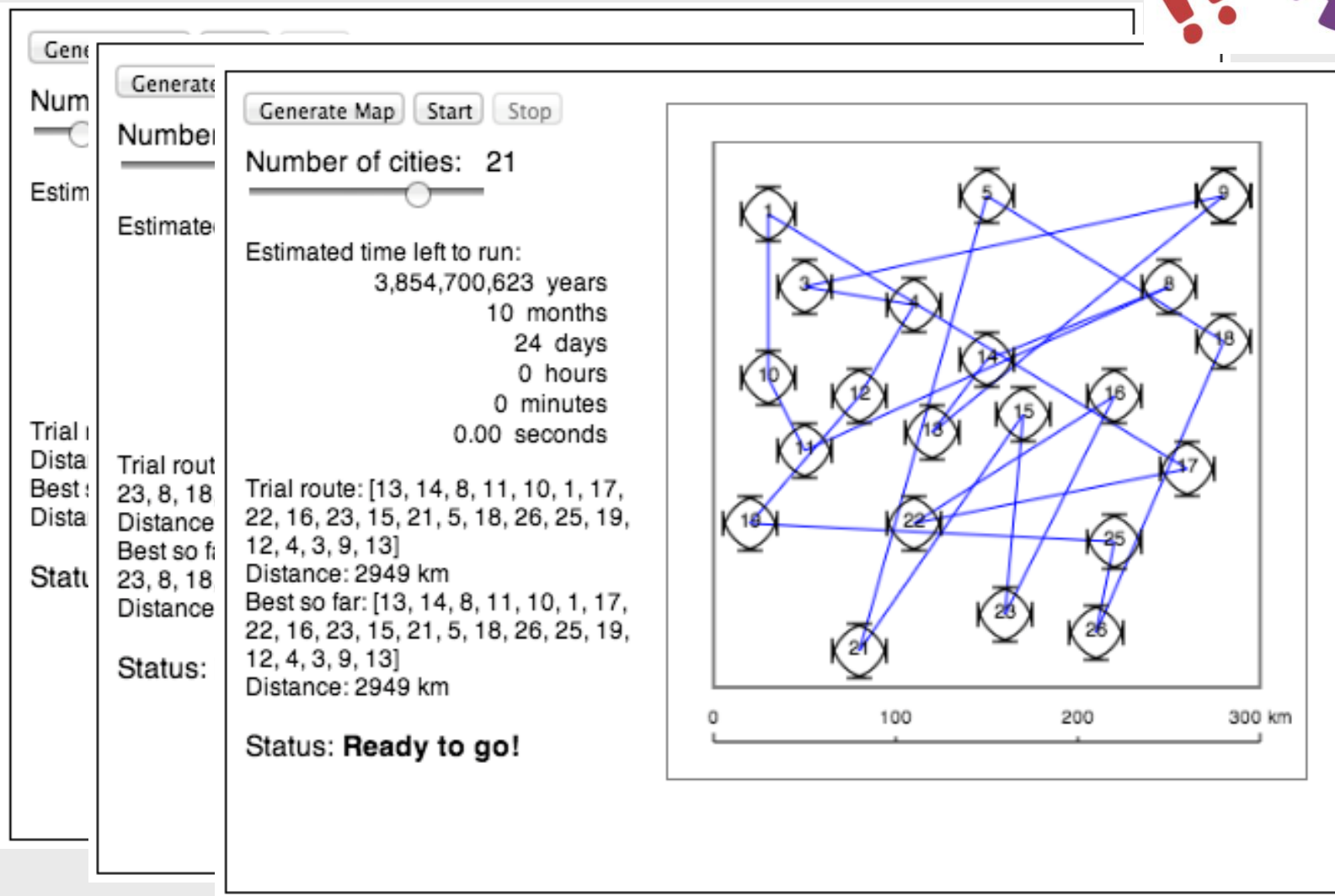
Undo Redo

Columns Hide All Routes

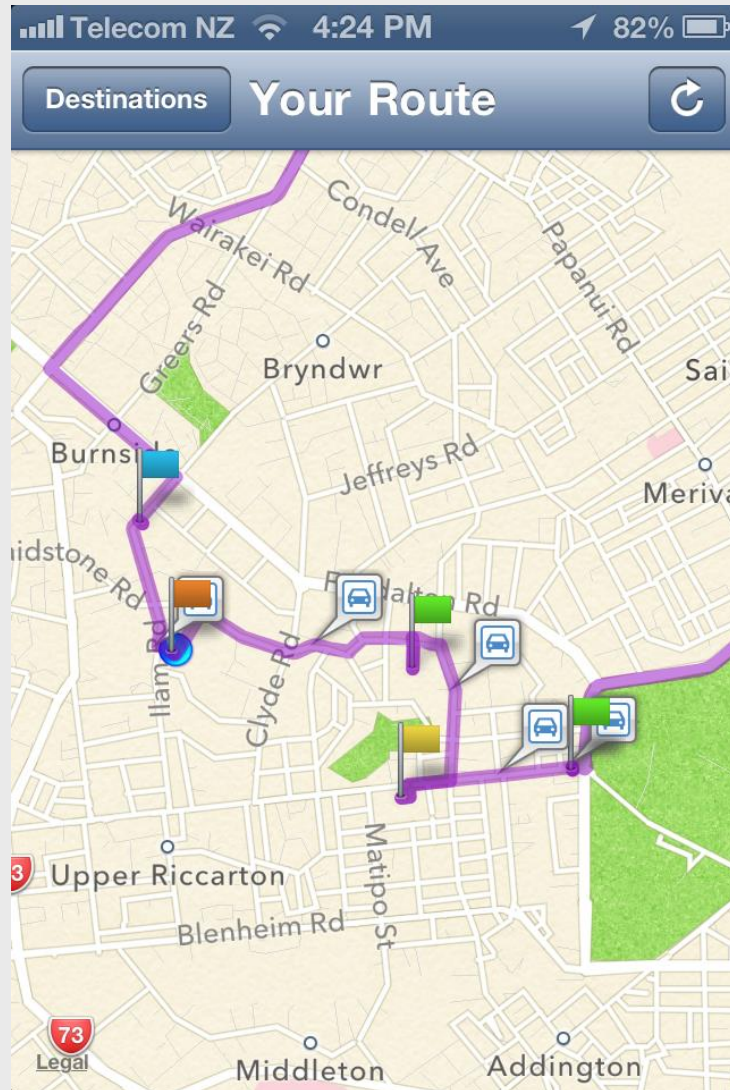
| Vehicle       | Day      | Stops   | Distance | Time        | Violation                           | Show                                | Optimize |
|---------------|----------|---------|----------|-------------|-------------------------------------|-------------------------------------|----------|
| EH-2          | Monday   | 15      | 149.3    | 661 mins    |                                     | <input type="checkbox"/>            | Optimize |
| EH-9          | Monday   | 13      | 260.2    | 694 mins    |                                     | <input checked="" type="checkbox"/> | Optimize |
| Customer      | Dist. Tr | Time Tr | Arrive   | Time On Str | Violations                          | Move/Refresh Ar                     |          |
| Customer 278  | 9.4      | 18 mins | 08:18    | 30 mins     |                                     |                                     |          |
| Customer 275  | 2.7      | 4 mins  | 08:52    | 30 mins     |                                     |                                     |          |
| Customer 3137 | 17.3     | 22 mins | 09:44    | 20 mins     |                                     |                                     |          |
| Customer 76   | 36.7     | 45 mins | 10:49    | 30 mins     |                                     |                                     |          |
| Customer 418  | 1.9      | 3 mins  | 11:22    | 30 mins     |                                     |                                     |          |
| Customer 397  | 0.0      | 0 mins  | 11:52    | 30 mins     |                                     |                                     |          |
| EH-5          | Monday   | 15      | 201.1    | 661 mins    | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | Optimize |
| EH-8          | Monday   | 11      | 316.3    | 690 mins    | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | Optimize |
| EH-10         | Monday   | 11      | 273.2    | 638 mins    | <input type="checkbox"/>            | <input type="checkbox"/>            | Optimize |
| EH-11         | Monday   | 13      | 269.5    | 732 mins    | <input type="checkbox"/>            | <input type="checkbox"/>            | Optimize |
| EH-6          | Monday   | 14      | 174.8    | 619 mins    | <input type="checkbox"/>            | <input type="checkbox"/>            | Optimize |
| EH-4          | Monday   | 14      | 154.7    | 645 mins    | <input type="checkbox"/>            | <input type="checkbox"/>            | Optimize |
| EH-7          | Monday   | 17      | 169.4    | 661 mins    | <input type="checkbox"/>            | <input type="checkbox"/>            | Optimize |
| EH-3          | Monday   | 13      | 210.6    | 668 mins    | <input type="checkbox"/>            | <input type="checkbox"/>            | Optimize |
| EH-1          | Monday   | 16      | 152.5    | 704 mins    | <input type="checkbox"/>            | <input type="checkbox"/>            | Optimize |



# Some computational problems **cannot be solved** by algorithm



# Eco maps



# Hamina, Finland



# Environmental impact

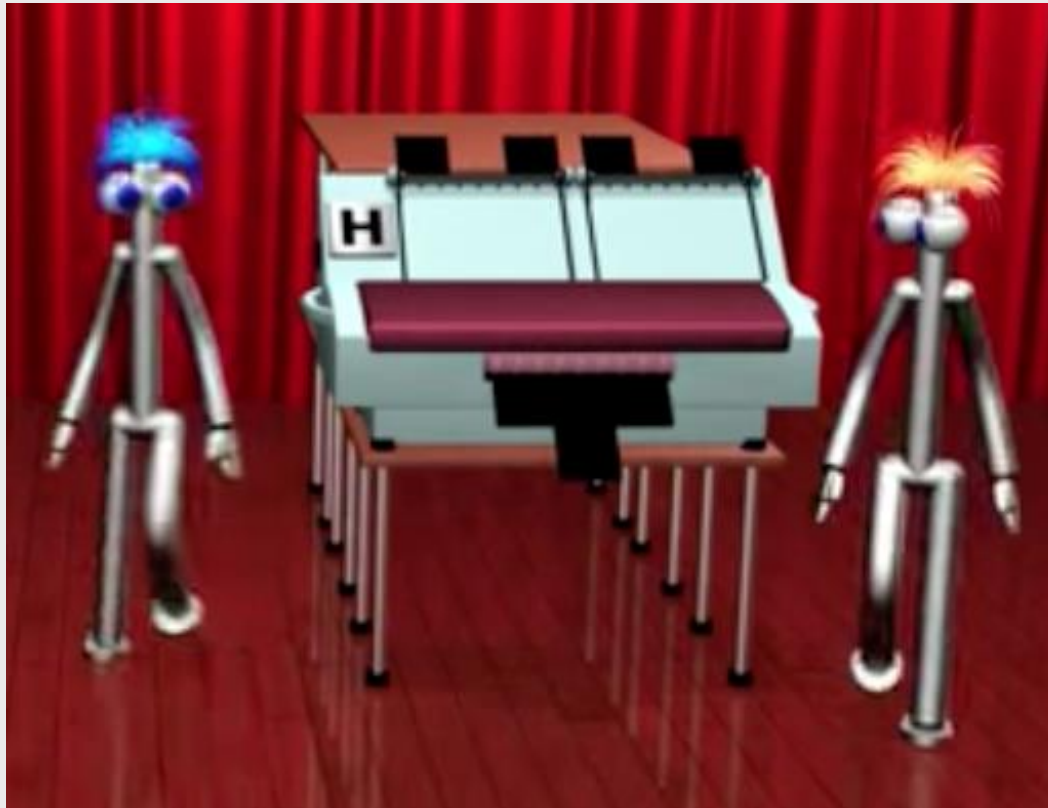
- **Speed of algorithm**
- **Structure of data**
- **Algorithms to optimise fuel consumption**



Some computational problems

# **cannot be solved** by algorithms

Proof That Computers Can't Do Everything  
(The Halting Problem)

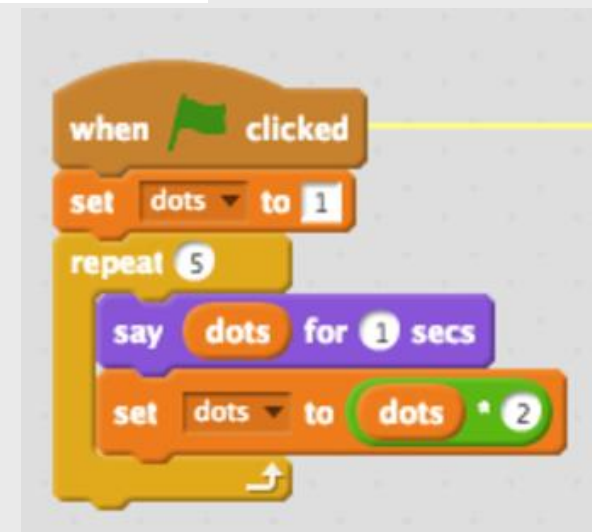
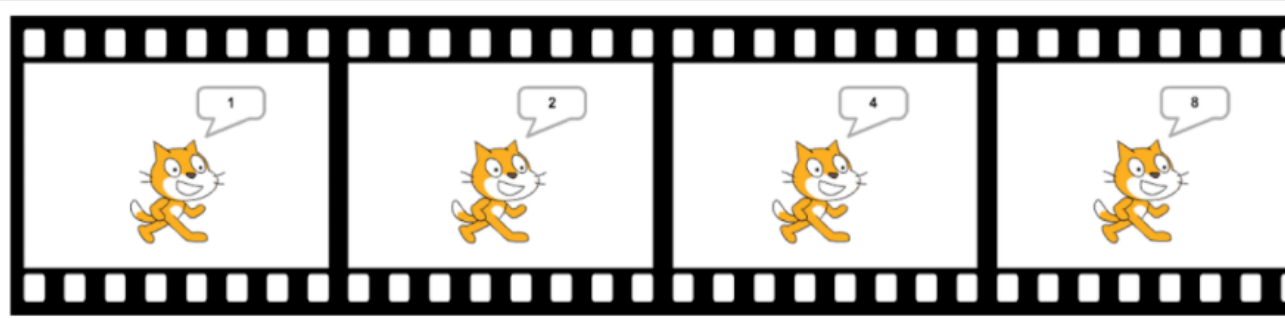
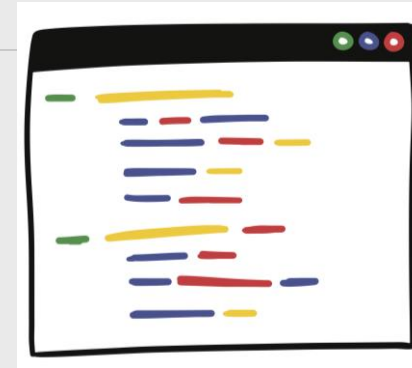


<https://www.youtube.com/watch?v=92WHN-pAFCs>

# Programs express algorithms and data

in a form that can be implemented on a computer.

- Plugging it in

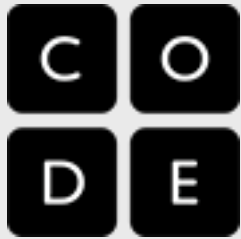


# “Coding”

codecademy



CODE CA AVENGERS



Hour of Code



[mothercoders]

things you love are  
Made w/ Code



# Kidbots



# Kidbots

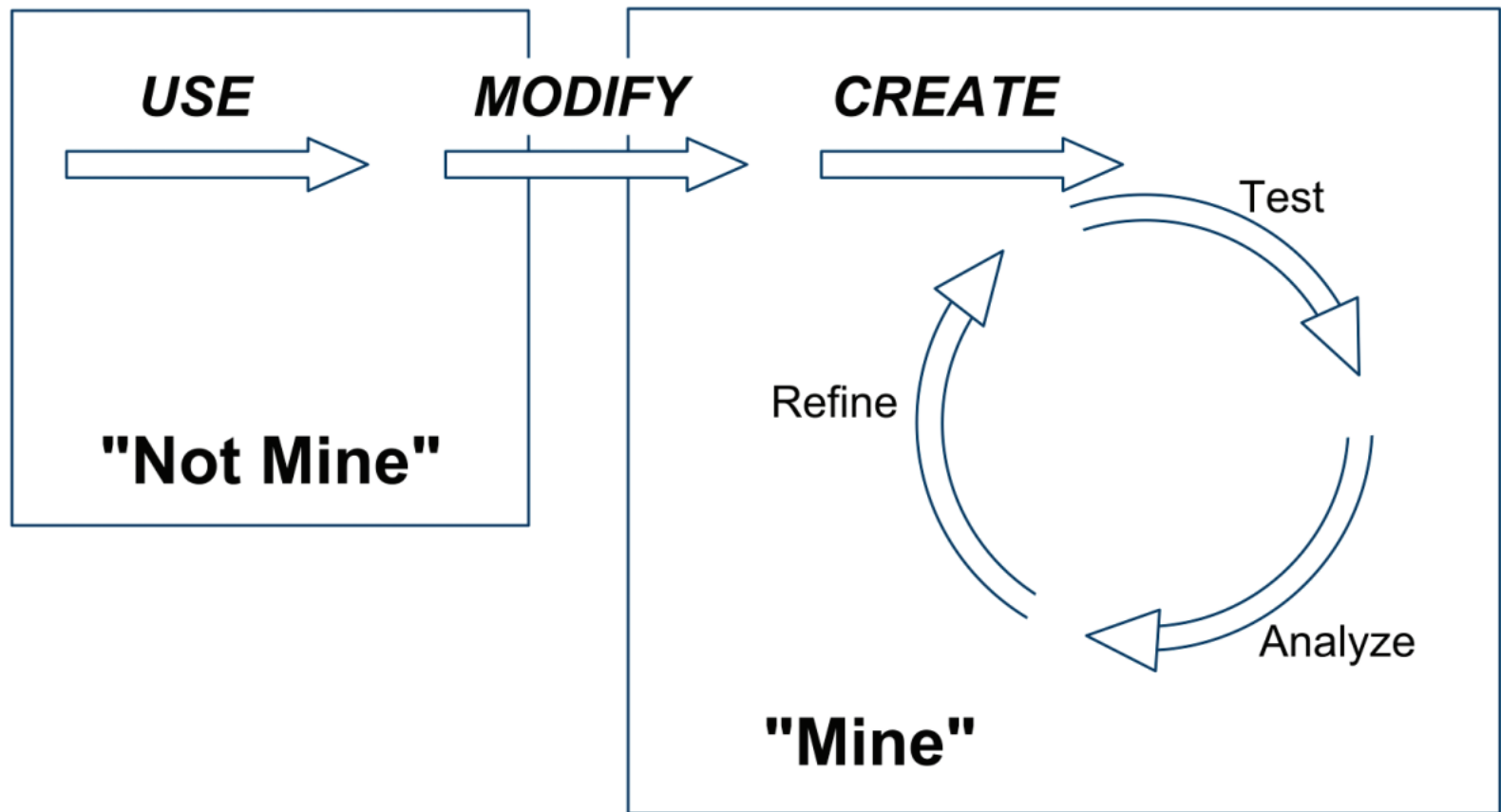


# Kidbots

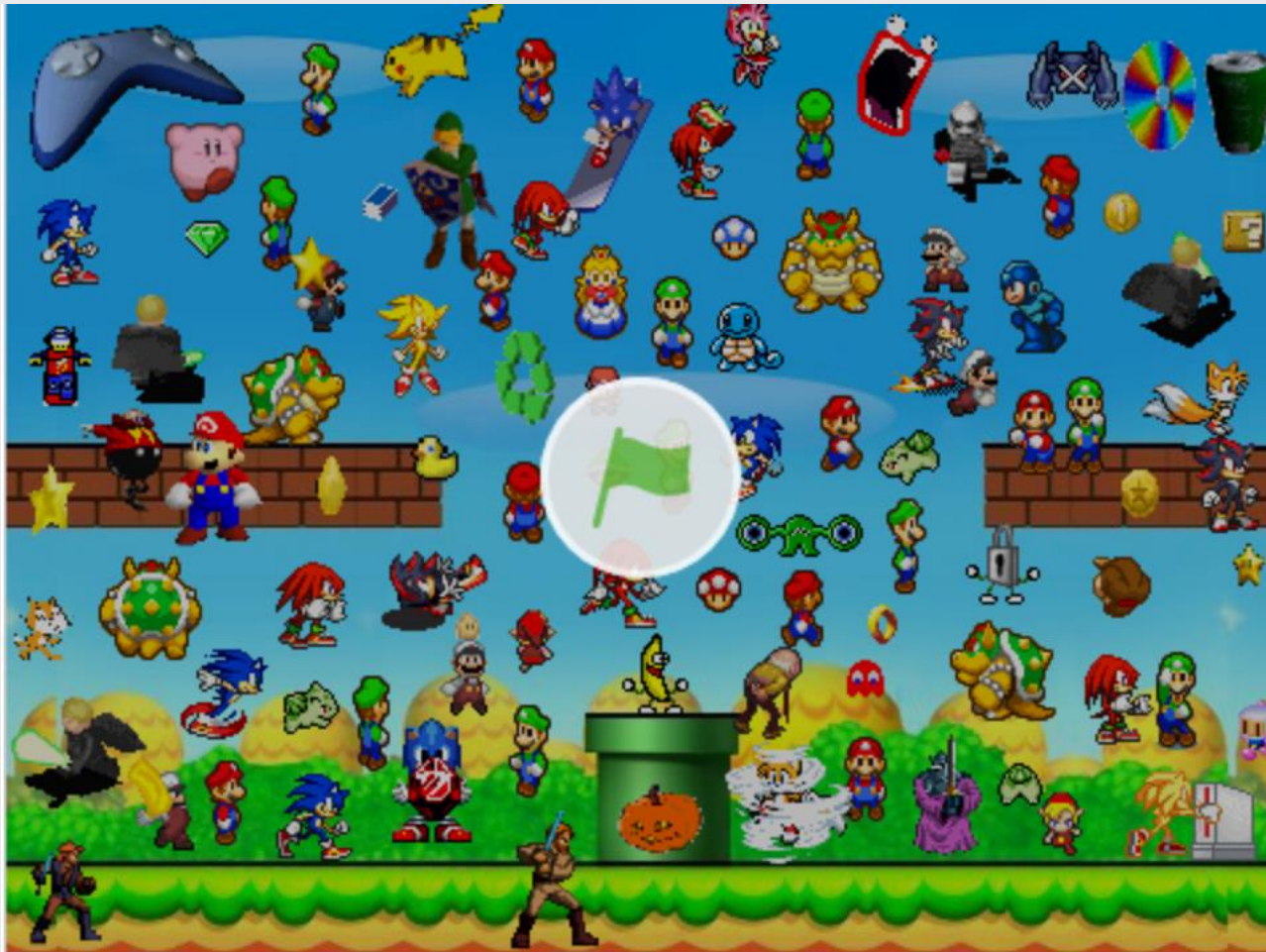


# Kidbots





# Spritefest

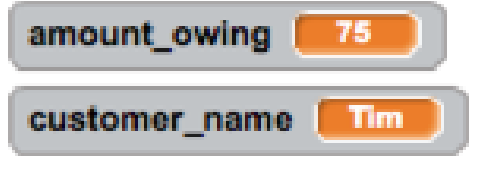


<http://scratch.mit.edu/projects/3169725/>

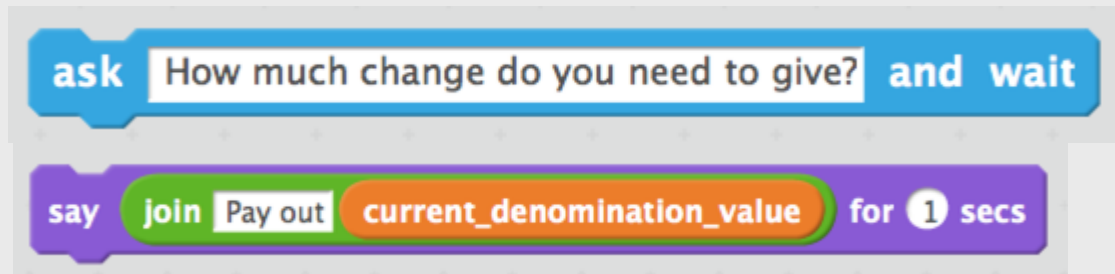
# Programs express algorithms and data

in a form that can be implemented on a computer.

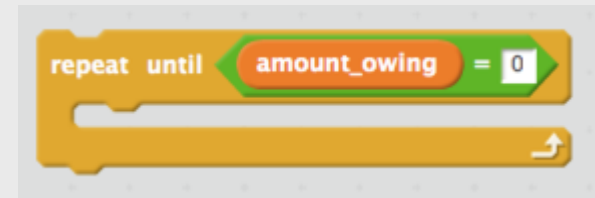
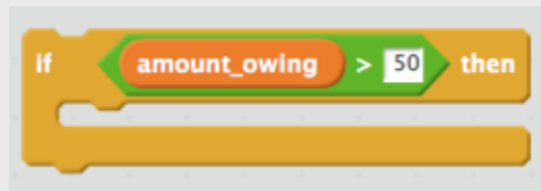
## Storage



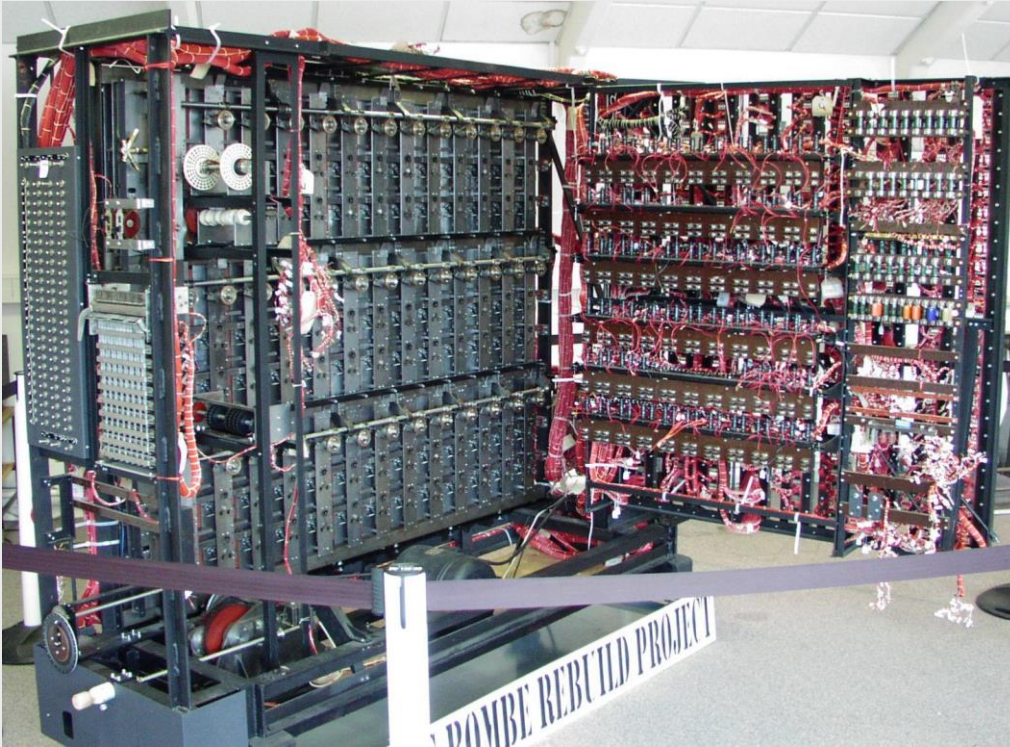
## Input/output



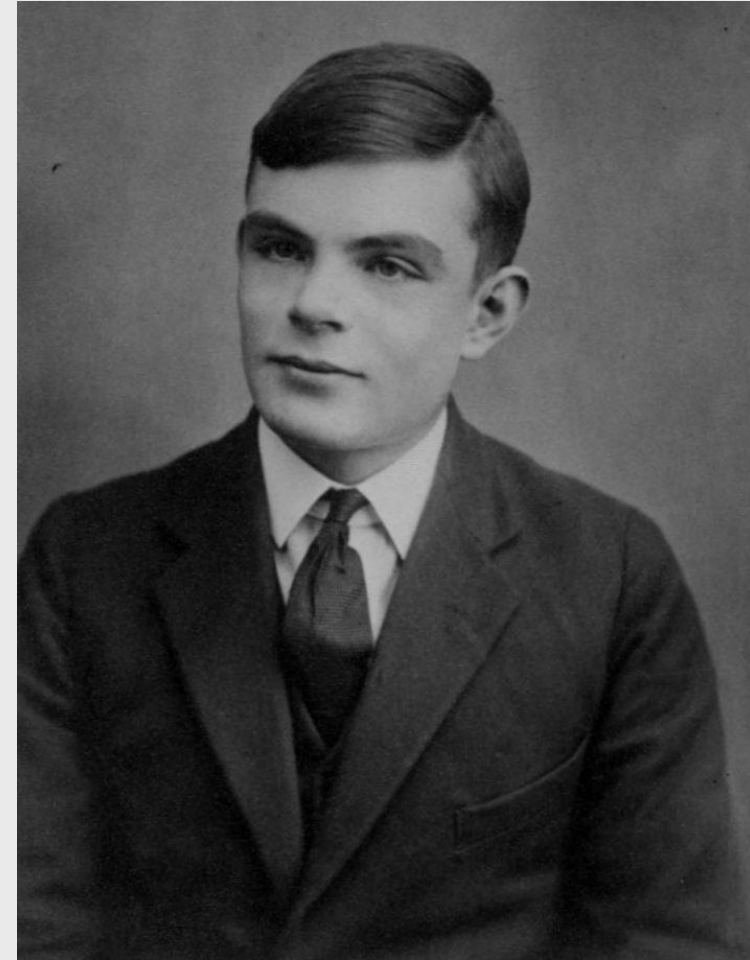
## Sequence, selection, iteration



# Turing-complete systems



<https://commons.wikimedia.org/wiki/File:Bombe-rebuild.jpg>



<http://www.turingarchive.org/viewer/?id=521&title=4>, Public Domain,  
<https://commons.wikimedia.org/w/index.php?curid=22828488>

# Computer programs are written for...

the next programmer

Many skills of a professional programmer are related to **social** context rather than the **technical** one.

```
def hs(s):  
    xxy = s[0]  
    for xxx in s:  
        if xxx > xxy:  
            xxy = xxx  
    return xxy
```

*Blackwell, A. (2002). What is programming? In 14th workshop of the Psychology of Programming Interest Group (pp. 204-218).*

```
def highest_score(scores):  
    """takes a list of scores and returns the highest one"""  
    highest_so_far = scores[0]  
    for score in scores:  
        if score > highest_so_far:  
            highest_so_far = score  
    return highest_so_far
```

# Elements of programming

■ **Analysis**

Communication

■ **Design**

Creativity

■ **Coding**

Skill

■ **Testing**

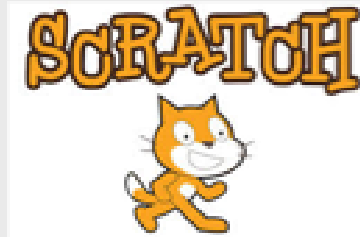
Rigour

■ **Debugging**

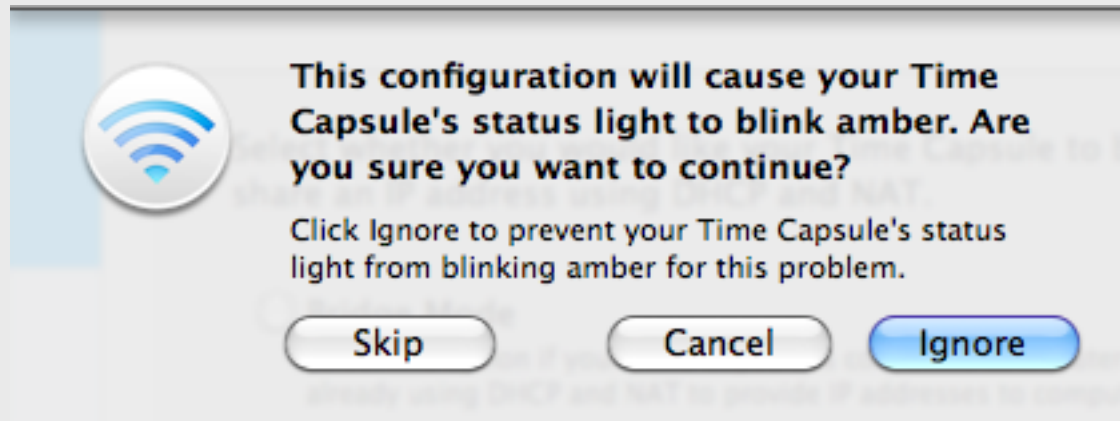
Persistence

# Choosing a language

- Good resources for teaching (online, books, peer support, software)
- Scaffolds from students' knowledge
- Encourages experimentation
- Motivates students
- Free



Computing systems are designed by  
**humans** to serve human ne



# Programs are written for...

humans

☒ I have read and accept the fare conditions and Conditions of Carriage  
(click box to confirm)

Start Over

Confirm







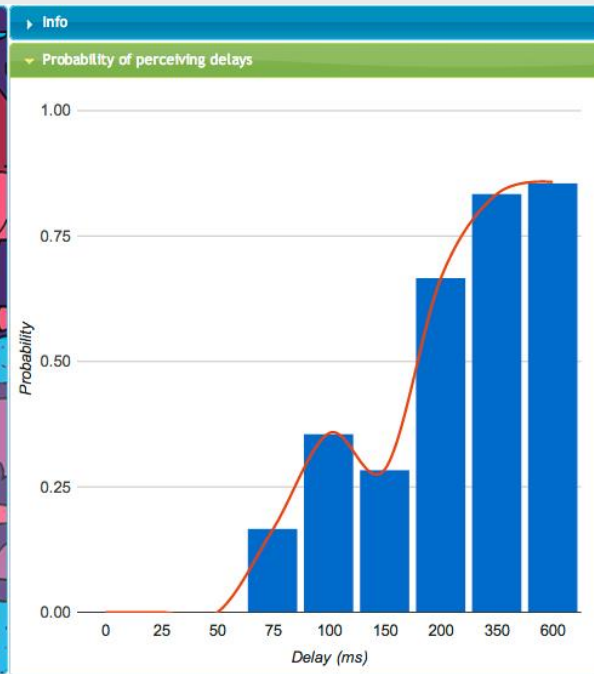


I have read and accept the fare conditions and Conditions of Carriage  
(click box to confirm)

**Start Over**

**Confirm**

# Reaction times?



# Computing systems are designed by **humans** to serve human needs



[https://www.youtube.com/watch?v=q\\_R9wQY4G5I](https://www.youtube.com/watch?v=q_R9wQY4G5I)

# Diversity – broadening participation



<http://techblog.nz/689-TheITGirls>



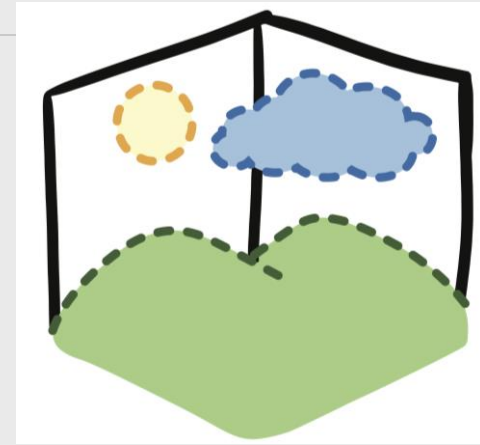
<http://au.tv.yahoo.com/sunrise/video/watch/21339850/blind-genius-building-apps-for-google/>

# A better motivation...

“The greatest tragedy I know of is that so many young people **never discover what they really want to do.**”

Edna Kerr  
(quoted by Dale Carnegie)

# Computing systems create **virtual** representations of natural and artificial phenomena



<https://www.kiwinet.org.nz/images/success-stories/AugmentedReality.jpg>

# Models include...

---

- **Simulations**
- **Spreadsheets**
- **Graphics**
- **AI**
- **Virtual machines**
- **Virtual/augmented reality**

# Protecting data and system resources

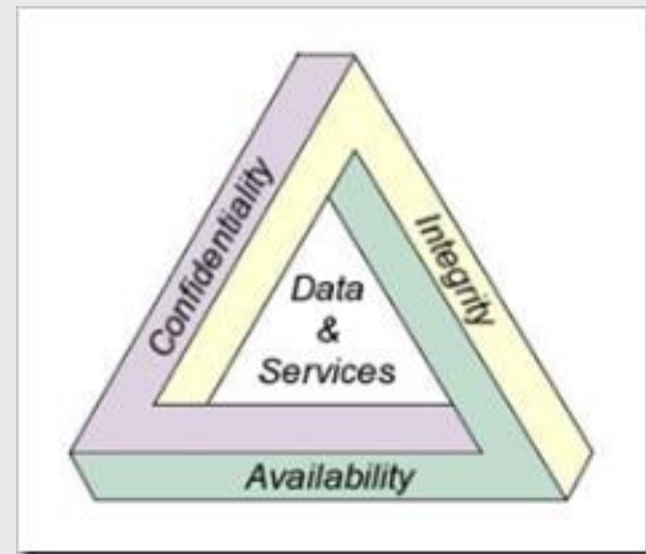
is critical in computing systems.

- Examples:

PKS

Hashed password

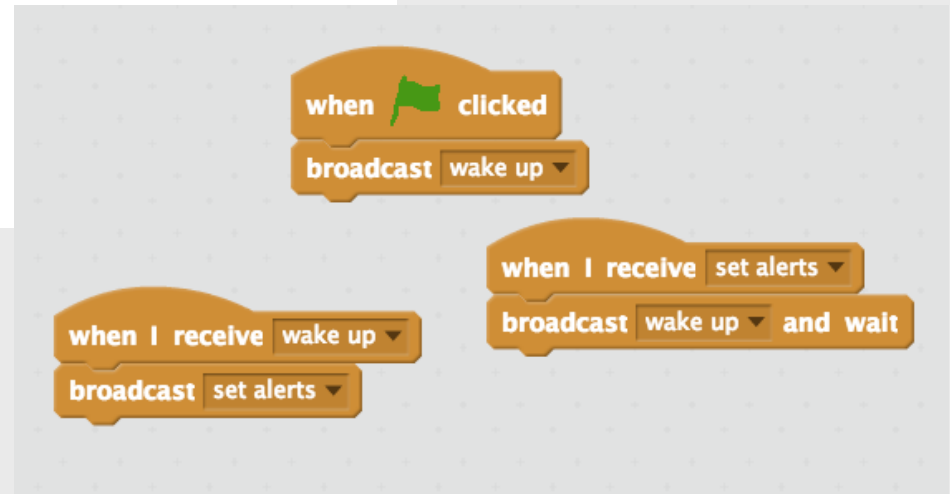
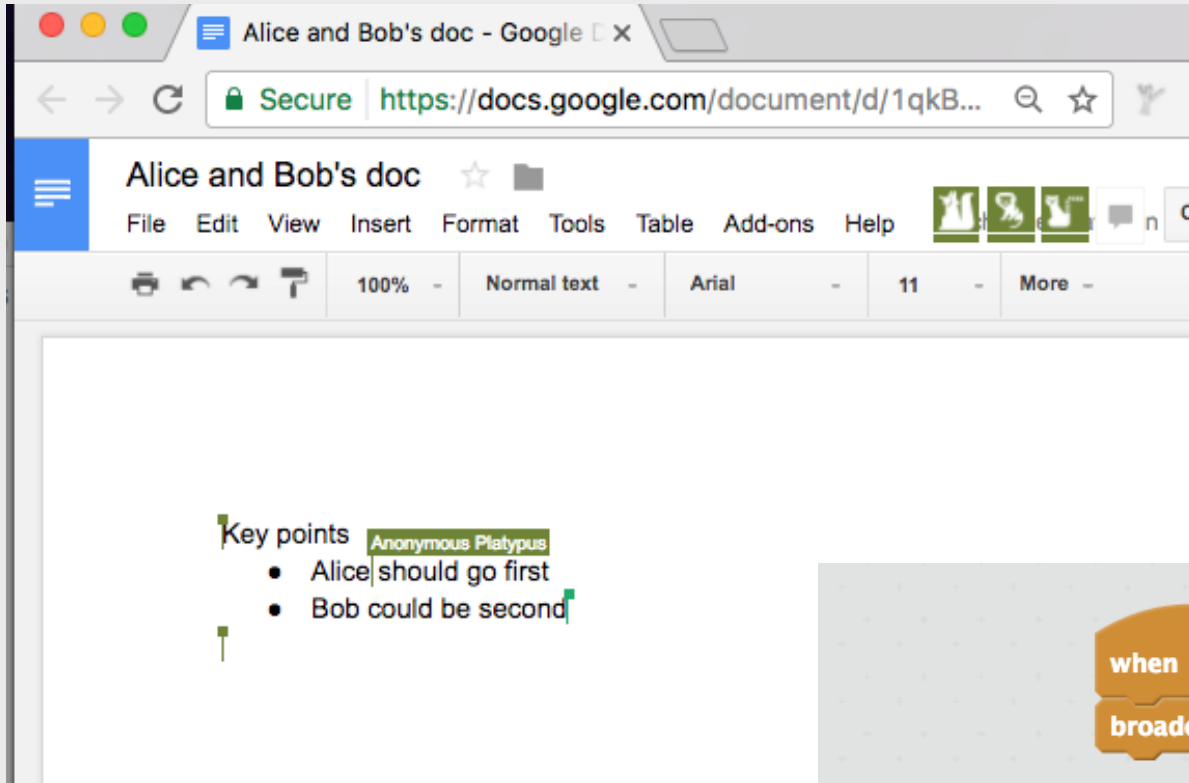
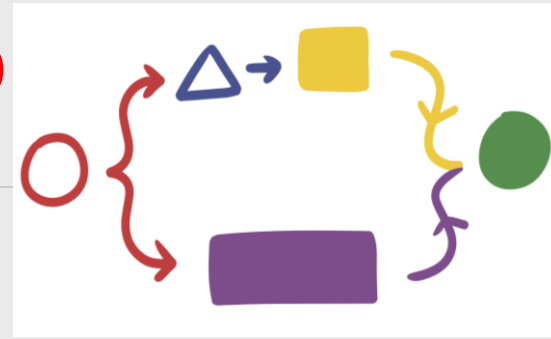
Coin toss



<https://blogs.technet.microsoft.com/seanearp/2007/08/01/layers-defense-in-depth-part-1/>

# Lockbox demo - PKS

# Time dependent operation in computing systems must be coordinated



# Parallel sorting network



# Integrated learning

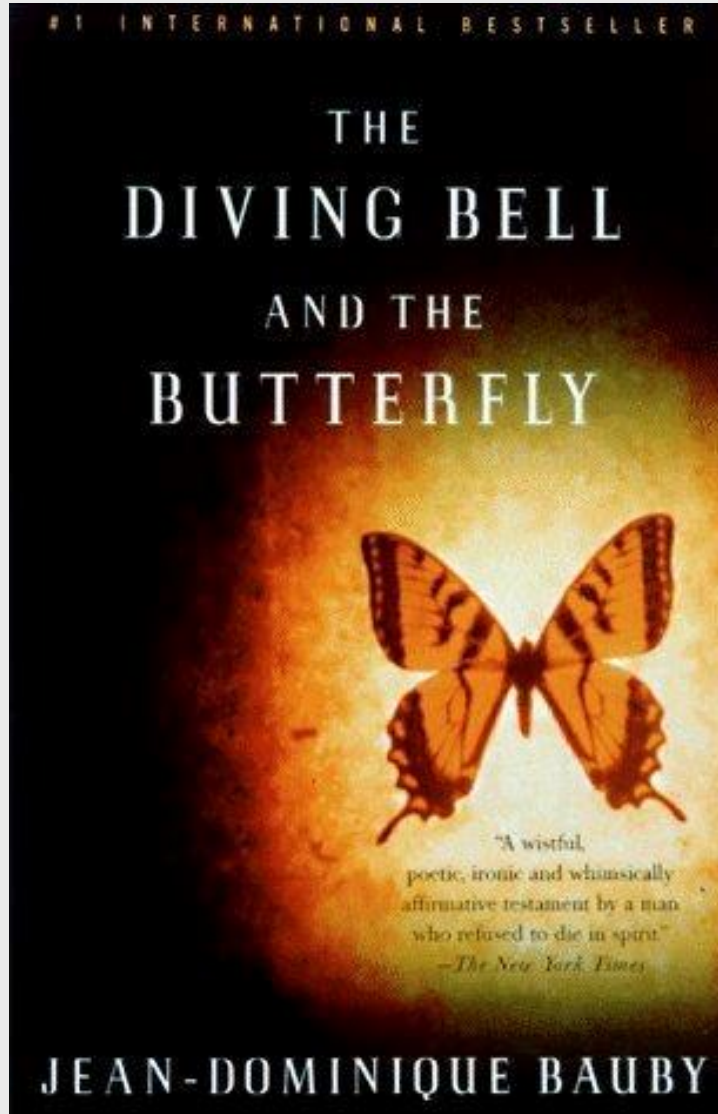


<http://chisnallwoodmusic.org.nz/wp/wp-content/uploads/2016/06/2016-06-17-16.16.27.mp4>

# Protocols facilitate communication between computing systems

| Name                                           | Meth... | Status | Type     | Initiator                                   | Size            | Time  | Waterfall |
|------------------------------------------------|---------|--------|----------|---------------------------------------------|-----------------|-------|-----------|
| www.google.com                                 | GET     | 302    | text/... | Other                                       | 433B            | 50ms  |           |
| ?gfe_rd=cr&ei=Sv62WNGgNtLr8AeJoKjoCA           | GET     | 200    | docu...  | <a href="https://www...">https://www...</a> | 38.2KB          | 263ms |           |
| gsa_super_g-64.gif                             | GET     | 200    | gif      | ?gfe_rd=cr...                               | 21.8KB          | 325ms |           |
| googlelogo_color_160x56dp.png                  | GET     | 200    | png      | ?gfe_rd=cr...                               | 7.0KB           | 181ms |           |
| rs=AA2YrTu-pFGahmg0_r6skyDfpReXWdGLng          | GET     | 200    | script   | ?gfe_rd=cr...                               | 20.9KB          | 50ms  |           |
| qi2_00ed8ca1.png                               | GET     | 200    | png      | ?gfe_rd=cr...                               | 3.3KB           | 48ms  |           |
| photo.jpg                                      | GET     | 200    | jpeg     | ?gfe_rd=cr...                               | (from disk c... | 2ms   |           |
| data:image/png;base...                         | GET     | 200    | png      | ?gfe_rd=cr...                               | (from mem...    | 0ms   |           |
| rs=ACT90oHm4z_AKMMfL6ZOBxtr6b2gaZSwyw          | GET     | 200    | script   | ?gfe_rd=cr...                               | 128KB           | 74ms  |           |
| rs=ACT90oHm4z_AKMMfL6ZOBxtr6b2gaZSwyw          | GET     | 200    | script   | rs=ACT90o...                                | 22.6KB          | 52ms  |           |
| search?client=mobile-gws-hp&hl=en-NZ&gs_rm=... | GET     | 200    | xhr      | rs=ACT90o...                                | 443B            | 219ms |           |
| nav_logo242.png                                | GET     | 200    | png      | Other                                       | (from disk c... | 2ms   |           |
| gen_204?s=mobilewebhp&atyp=csi&ei=S_62WK...    | GET     | 204    | text/... | Other                                       | 23B             | 193ms |           |
| rs=AA2YrTvYlgFmYK0OmsJFhJXcUYROuDYVA           | GET     | 200    | style... | ?gfe_rd=cr...                               | 2.7KB           | 55ms  |           |
| gen_204?atyp=i&ct=14&cad=i&ogsr=1&id=19001...  | GET     | 204    | text/... | Other                                       | 23B             | 185ms |           |
| count                                          | POST    | 200    | xhr      | rs=AA2YrTu...                               | 570B            | 539ms |           |
| gen_204?atyp=i&ct=14&cad=v&ogsr=1&id=1900...   | GET     | 204    | text/... | Other                                       | 46B             | 185ms |           |

# Locked-in syndrome



<https://teachinglondoncomputing.org/resources/inspiring-unplugged-classroom-activities/the-locked-in-activity/>

# The big ideas

1. **Data** is represented in digital form on a computer.
2. **Algorithms** interact with data to solve computational problems.
3. The **performance** of algorithms can be modelled and evaluated.
4. Some computational problems **cannot be solved** by algorithms.
5. **Programs** express algorithms and data in a form that can be implemented on a computer.
6. Computing systems are designed by **humans** to serve human needs.
7. Computing systems create **virtual** representations of natural and artificial phenomena.
8. **Protecting** data and system resources is critical in computing systems.
9. **Time dependent** operations in computing systems must be coordinated.
10. **Protocols** facilitate communication between computing systems.

# Digital technologies is about... people

- Communication
- Information
- Interfaces
- Privacy
- Safety
- Solving real problems
- Careers



[www.orionhealth.com](http://www.orionhealth.com)



He aha te mea nui ki tēnei ao? Māku e ki atu. He tangata, he tangata, he tangata.

[www.youtube.com/watch?v=cdgQpa1pUUE](http://www.youtube.com/watch?v=cdgQpa1pUUE)

# Computer Science without a computer

CS Unplugged is a collection of free teaching material that teaches Computer Science through engaging games and puzzles that use cards, string, crayons and lots of running around.

Welcome to the new CS Unplugged!

This updated website has unit plans, lesson plans, teaching videos, curriculum integration activities, and programming exercises to plug in the Computer Science concepts they have just learnt unplugged.

The original activities are still available at [classic.csunplugged.org](http://classic.csunplugged.org).



## What is Computer Science?

### Topics

### Curriculum Integrations



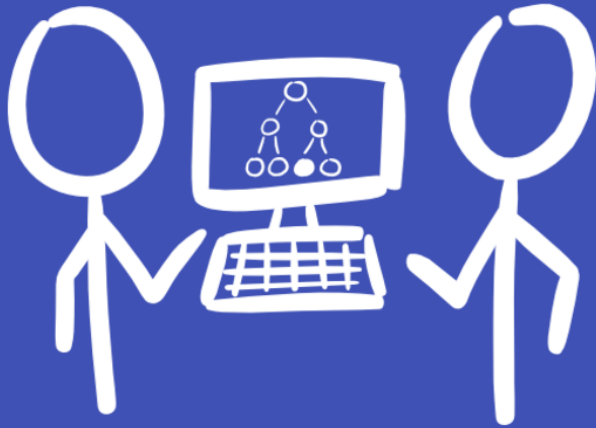
## How do I teach CS Unplugged?

### Printables

### Classic CS Unplugged

# Department of Fun Stuff

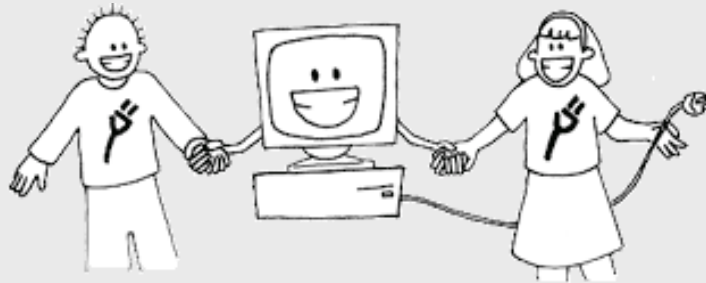




## Computer Science Field Guide

- Algorithms
- Programming Languages
- Human Computer Interaction
- Data Representation
- Coding - Introduction
- Coding - Compression
- Coding - Encryption
- Coding - Error Control
- Artificial Intelligence
- Complexity and tractability
- Computer Graphics
- Computer Vision
- Formal Languages
- Network Communication Protocols
- Software Engineering

**[http://www.cosc.canterbury.ac.nz/  
research/RG/CSE/big-ideas/](http://www.cosc.canterbury.ac.nz/research/RG/CSE/big-ideas/)**



**tim.bell@canterbury.ac.nz**