Google Computer Science for Schools

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

FACULTY OF ENGINEERING AND BUILT ENVIRONMENT

www.newcastle.edu.au

FACULTY OF EDUCATION AND ARTS

www.newcastle.edu.au

Computer Science… unplugged!

Dr Elena Prieto

School of Education

Faculty of Education and Arts

18th July 2016

Why unplugged?

- As you know, Computer Science is not just coding!
- Computational thinking and algorithmic thinking are at its core and many of its principles can be taught without the aid of a computer
- The activities this morning are designed for 'rainy days' when things may go wrong and computers are not available
  - And yet they are fundamental at the high end of program design

Presentation Content

- The problems
- Problem solving (plugged)
- Problem solving (unplugged)
- The problems (revisited)

The problems

1. The RTA want to make sure that they connect all parks by at least one super-fast road, but they have a very low budget for it. The company that is awarded the contract has to choose the roads that will connect the parks in Newcastle, for the cheapest amount.
2. Council wants to set up a system of security cameras such that for every road connecting two parks at least one of their endpoints has a camera. Because their budget is also low, they would like the number of cameras to be the minimum possible.
3. The NSW SES wants to establish a system of specialised fire trucks for the parks in Newcastle so that each park either has one of these new trucks, or is linked to it by a super-fast road.

Problem Solving (plugged)

- **Analyse** the problem to understand how to approach solving it
- Use methods to **model** the problem, using diagrams or graphs
- Adapt an existing **algorithm** for solving a similar problem to the given problem, or design a new algorithm
- Write out the algorithm in **pseudocode** and step through it using test case data to verify it makes sense
- **Code** the algorithm in a chosen language using the pseudo-code
- Verify the algorithm works as expected, by **testing** it against expected input and output

Problem Solving (plugged)

Analyse

model

algorithm

pseudocode

Code

testing

Problem Solving (plugged)

- Analyse
- Model
- Algorithm
- Pseudocode
- Code
- Test

Problem Solving (unplugged)

- Analyse
- Model
- Algorithm
- Pseudocode
- Run by hand!!!

Analyse…

1. The RTA wants to make sure that they connect all **parks** by at least one super-fast **road**, but they have a very low budget for it. The company that is awarded the contract has to choose the roads that will connect the parks in Newcastle, for the cheapest amount.

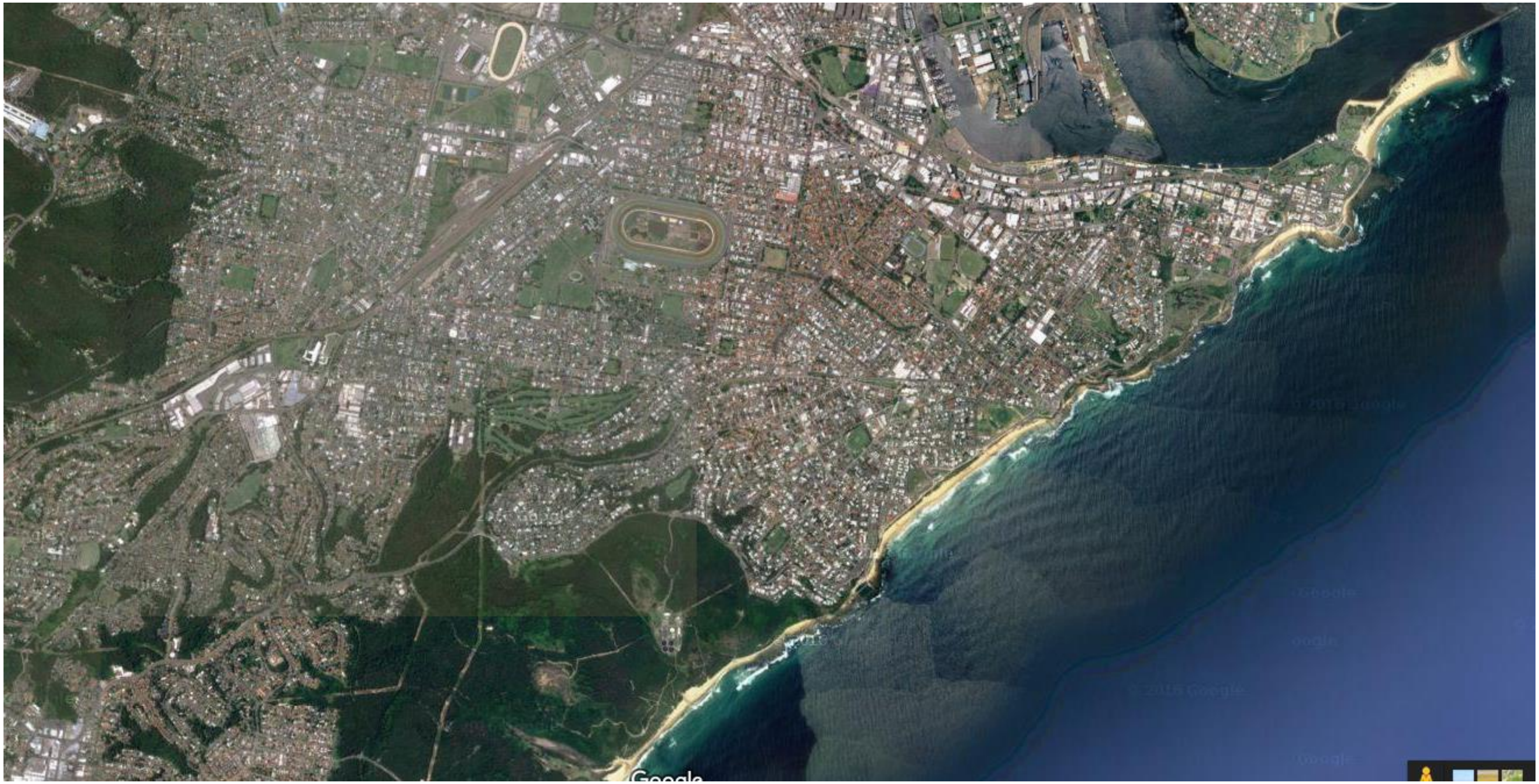
2. Council wants to set up a system of security cameras such that for every **road** connecting two **parks** at least one of their endpoints has a camera. Because their budget is also low, they would like the number of cameras to be the minimum possible.

3. The NSW SES wants to establish a system of specialised fire trucks for the **parks** in Newcastle so that each park either has one of these new trucks, or is linked to it by a super-fast **road**.

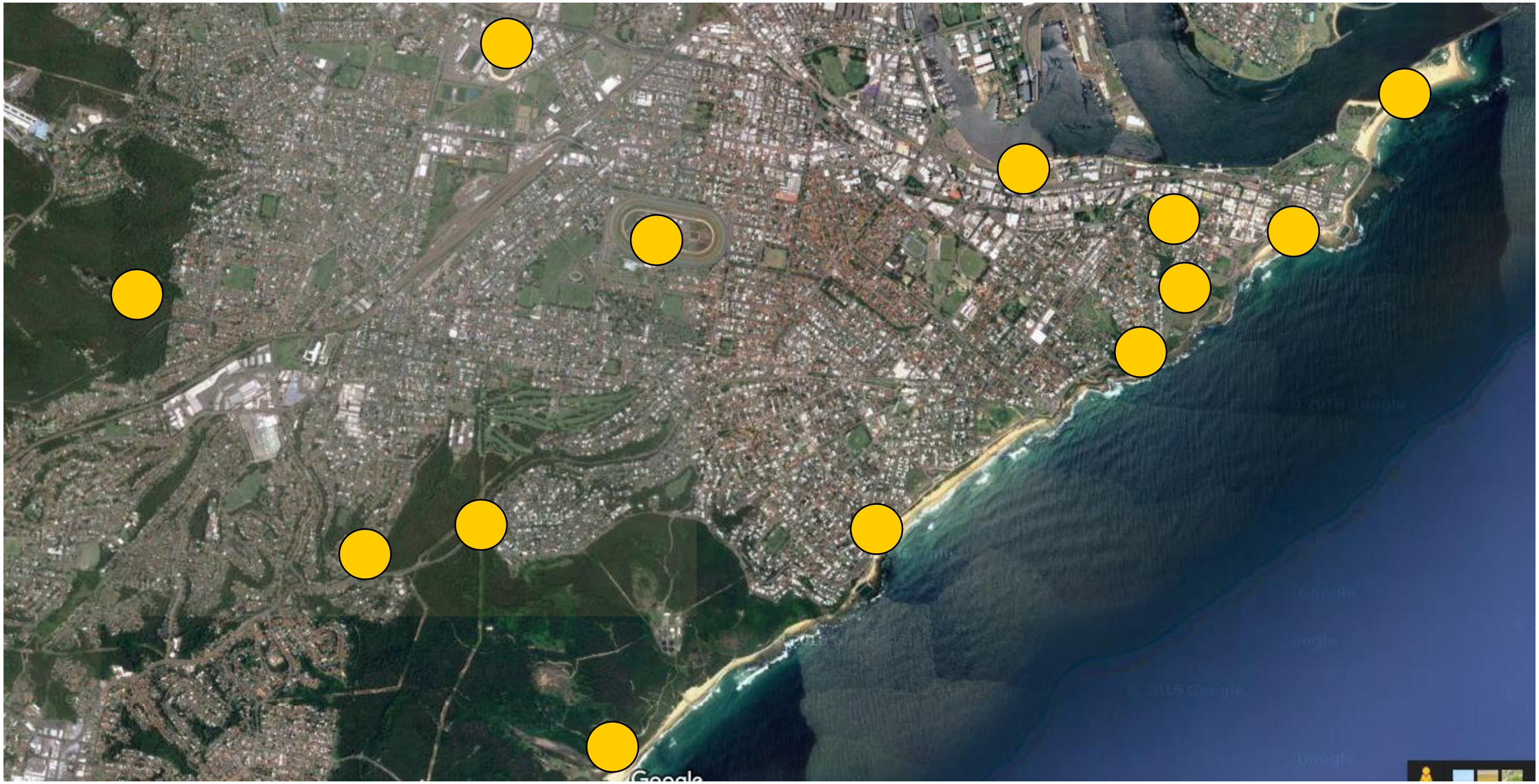
Problem Solving (unplugged)

- Analyse
- **Model**
- Algorithm
- Pseudocode
- Run by hand

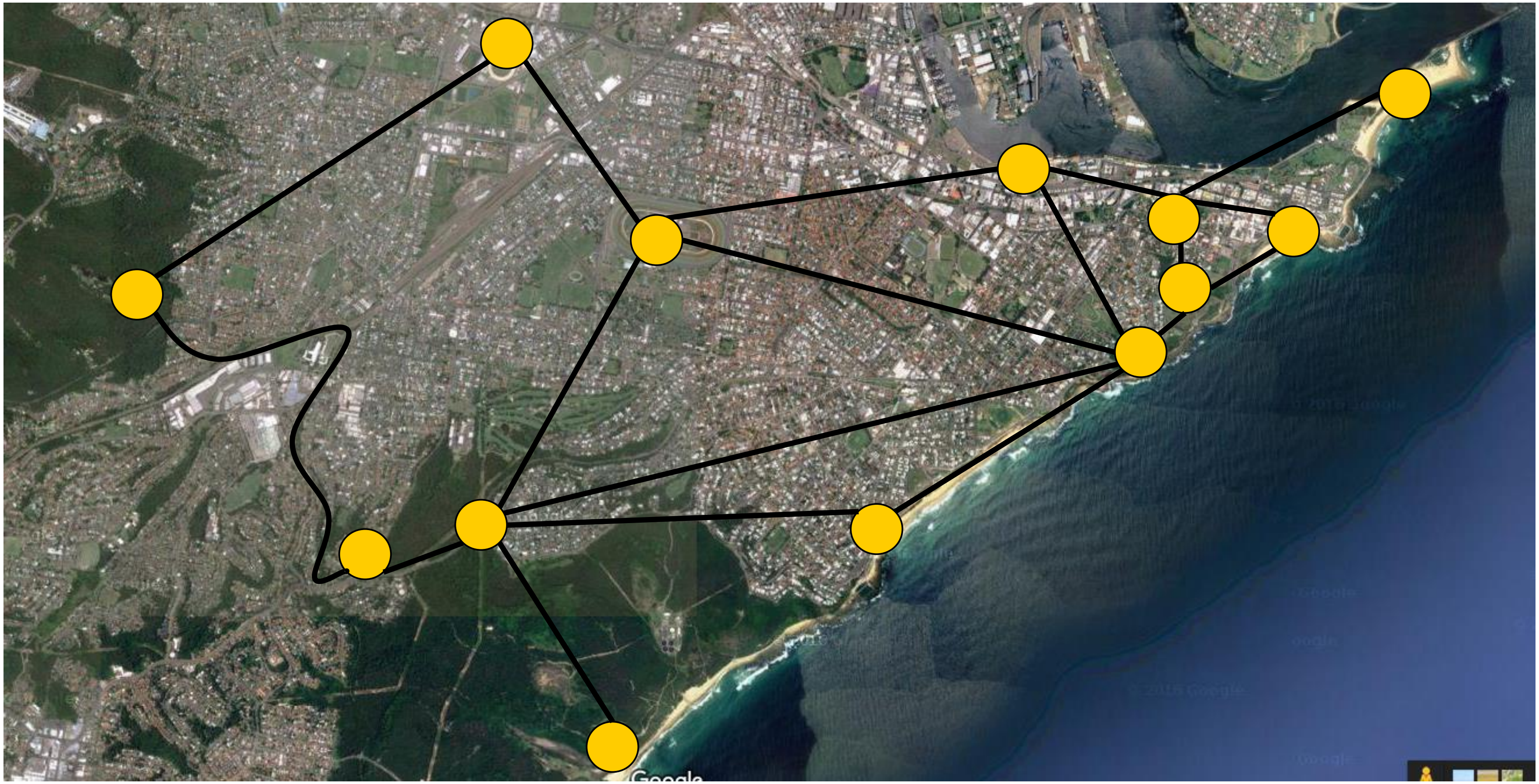
Model…



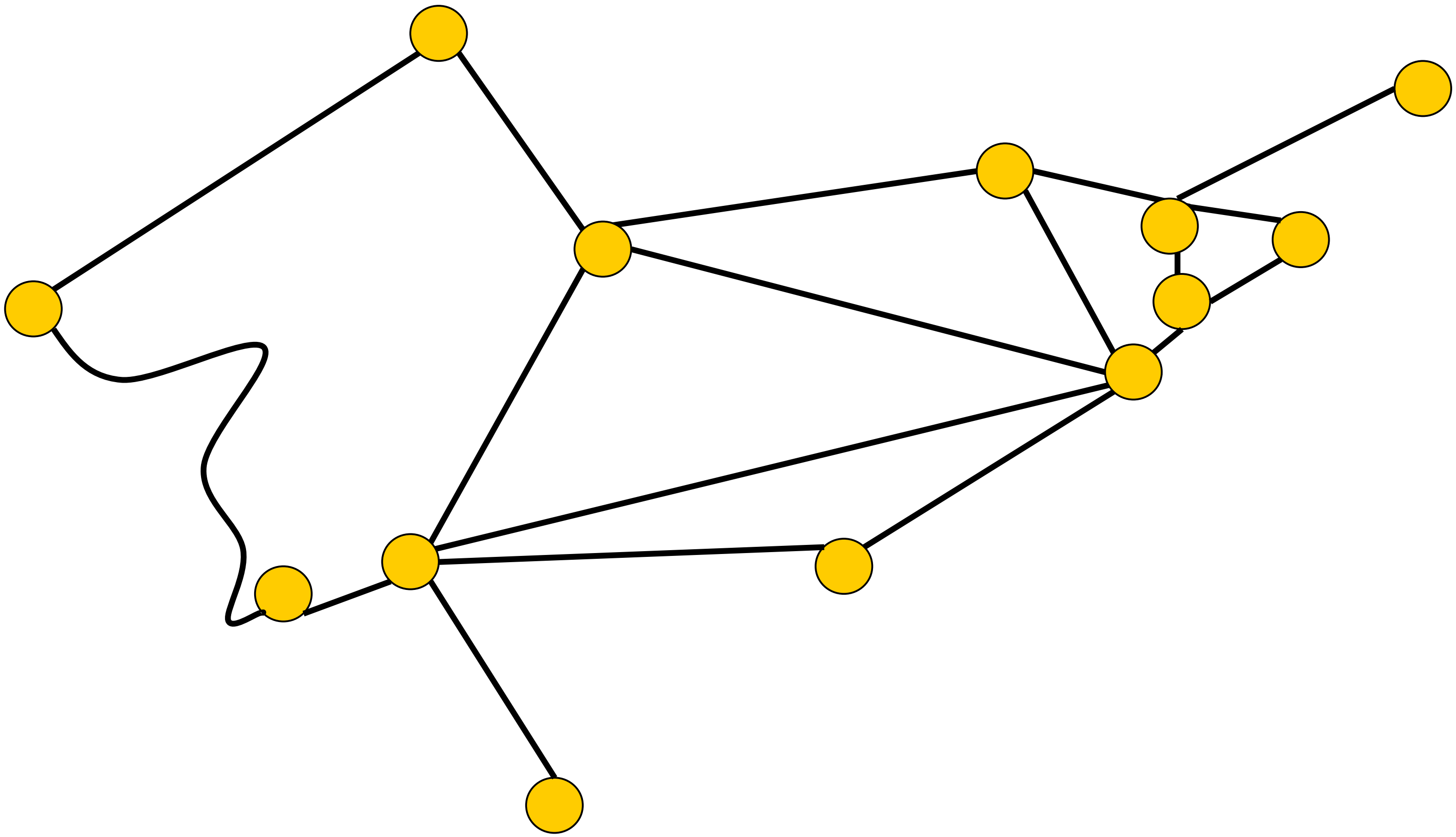
Model…



Model…

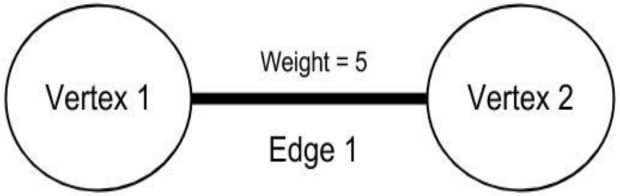


Model…

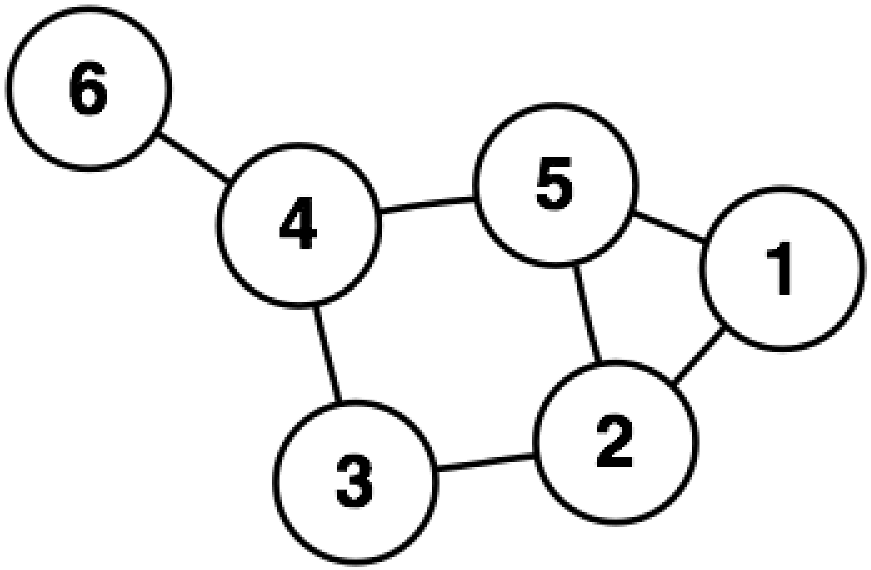


Graphs

- Graphs have Vertices (plural of Vertex)
- Edges that connect them
- Graphs can be weighted - Edges have weights, such as a distance or cost



Graphs



V:={1,2,3,4,5,6}

E:={{1,2},{1,5},{2,3},{2,5},{3,4},{4,5},{4,6}}

Problem Solving (unplugged)

- Analyse
- Model
- **Algorithm**
- Pseudocode
- Run by hand

Algorithm…

1. The RTA wants to make sure that they connect all **parks** by at least one super-fast **road**, but they have a very low budget for it. The company that is awarded the contract has to choose the roads that will connect the parks in Newcastle, for the cheapest amount.

2. Council wants to set up a system of security cameras such that for every **road** connecting two **parks** at least one of their endpoints has a camera. Because their budget is also low, they would like the number of cameras to be the minimum possible.

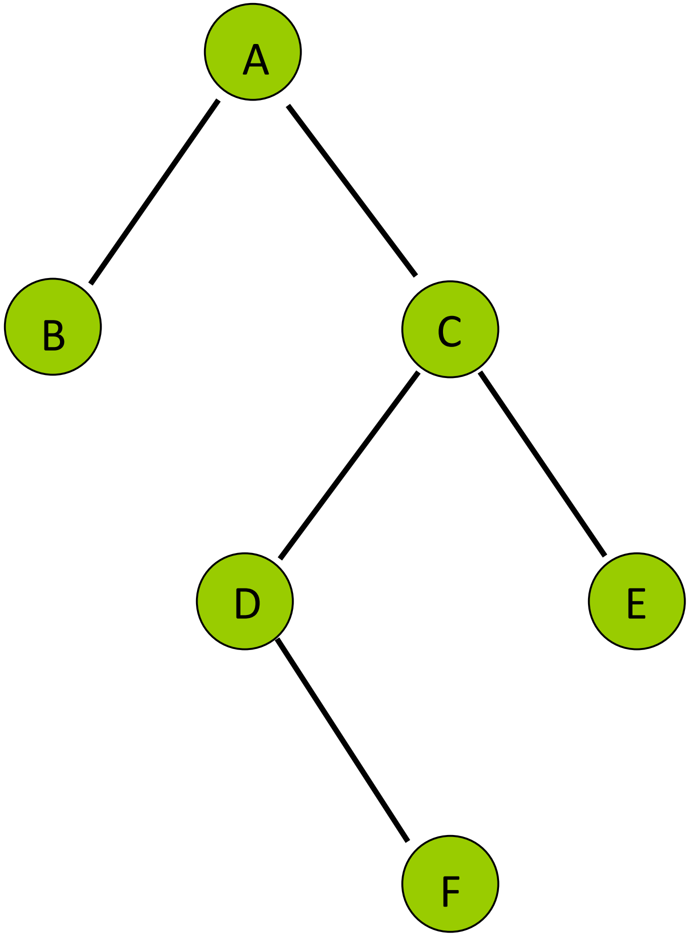
3. The NSW SES wants to establish a system of specialised fire trucks for the **parks** in Newcastle so that each park either has one of these new trucks, or is linked to it by a super-fast **road**.

Algorithm…

1. The RTA wants to make sure that they connect all **parks** by at least one super-fast **road**, but they have a very low budget for it. The company that is awarded the contract has to choose the roads that will connect the parks in Newcastle, for the cheapest amount.

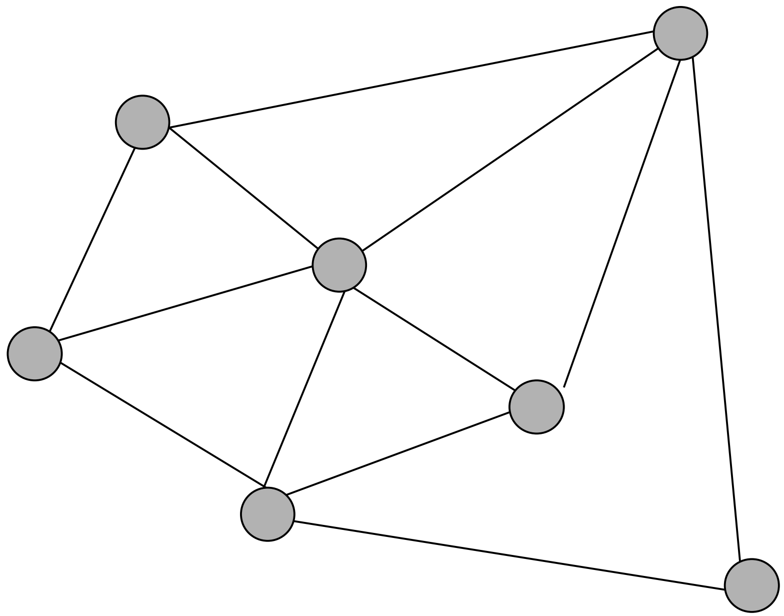
Model a little more…

- A tree is a graph that looks like this:



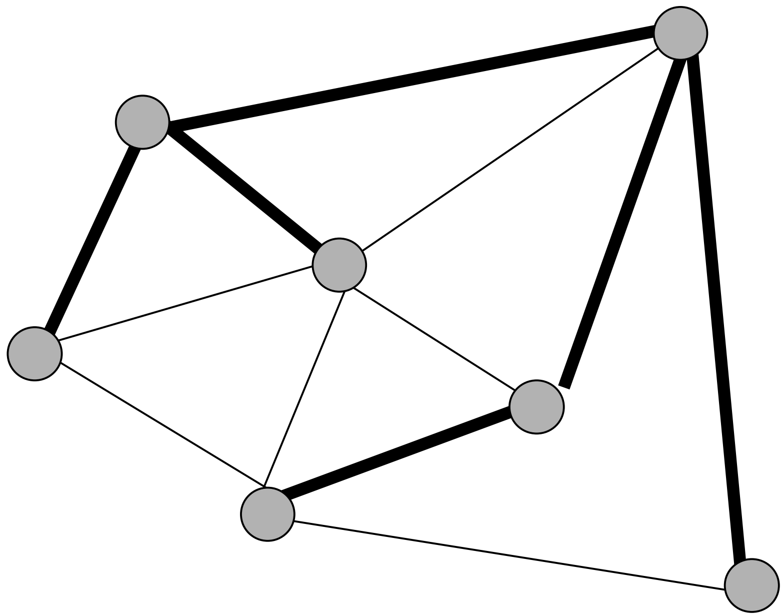
Minimum Spanning Trees

- A SPANNING TREE is a tree 'living' inside a graph containing all its vertices.



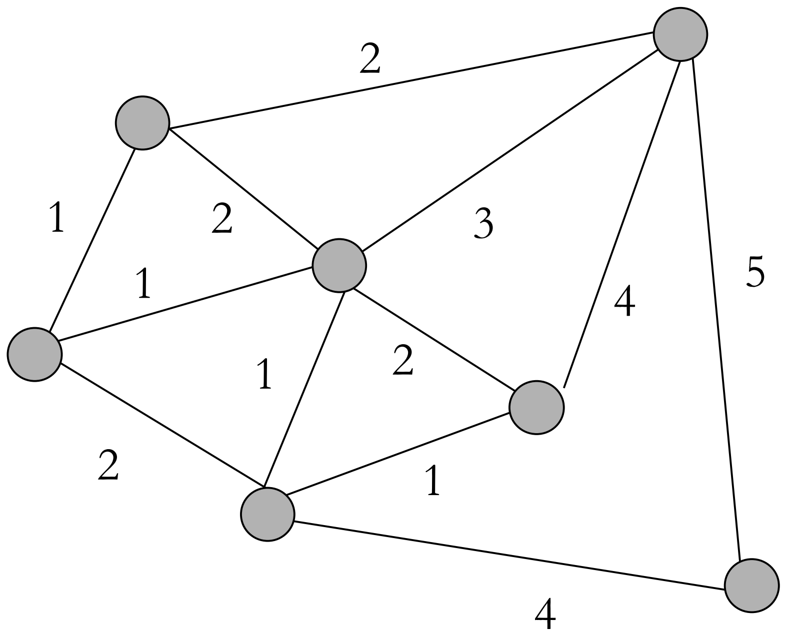
Minimum Spanning Trees

- A SPANNING TREE is a tree 'living' inside a graph containing all its vertices.



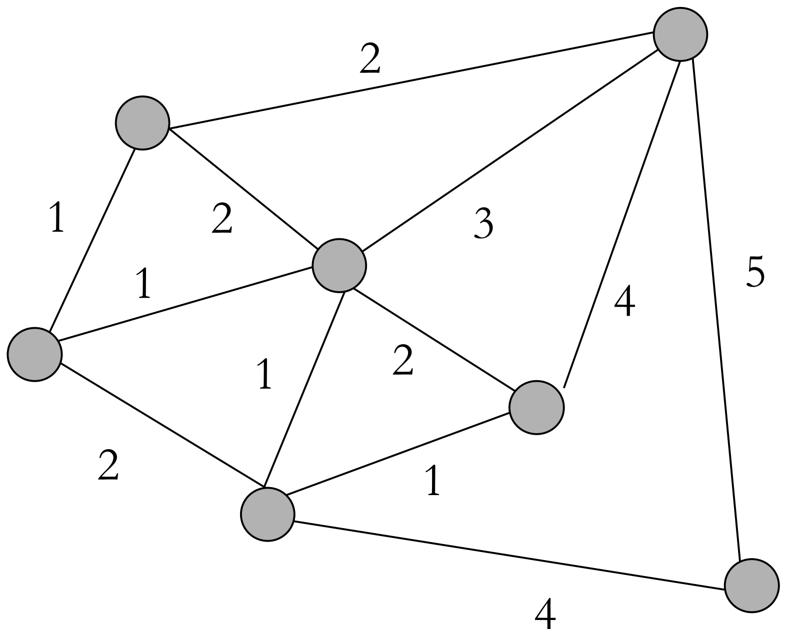
Minimum Spanning Trees

- But what happens if the edges in the tree have weights?



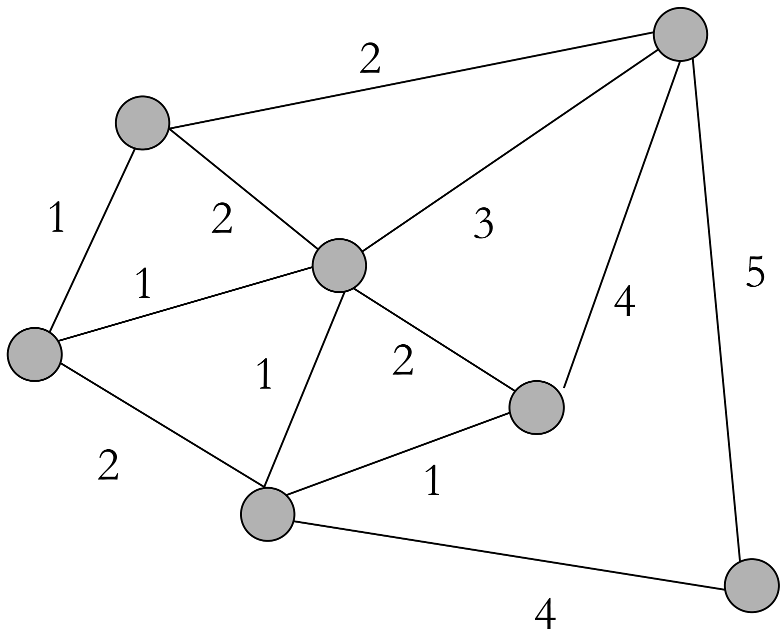
Minimum Spanning Trees

- But what happens if the edges in the tree have weights?



Like the super-fast roads we have to build on a low budget in our first problem!

Minimum Spanning Tree



Problem Solving (unplugged)

- Analyse
- Model
- Algorithm
- **Pseudocode**
- Run by hand

Pseudocode for MST

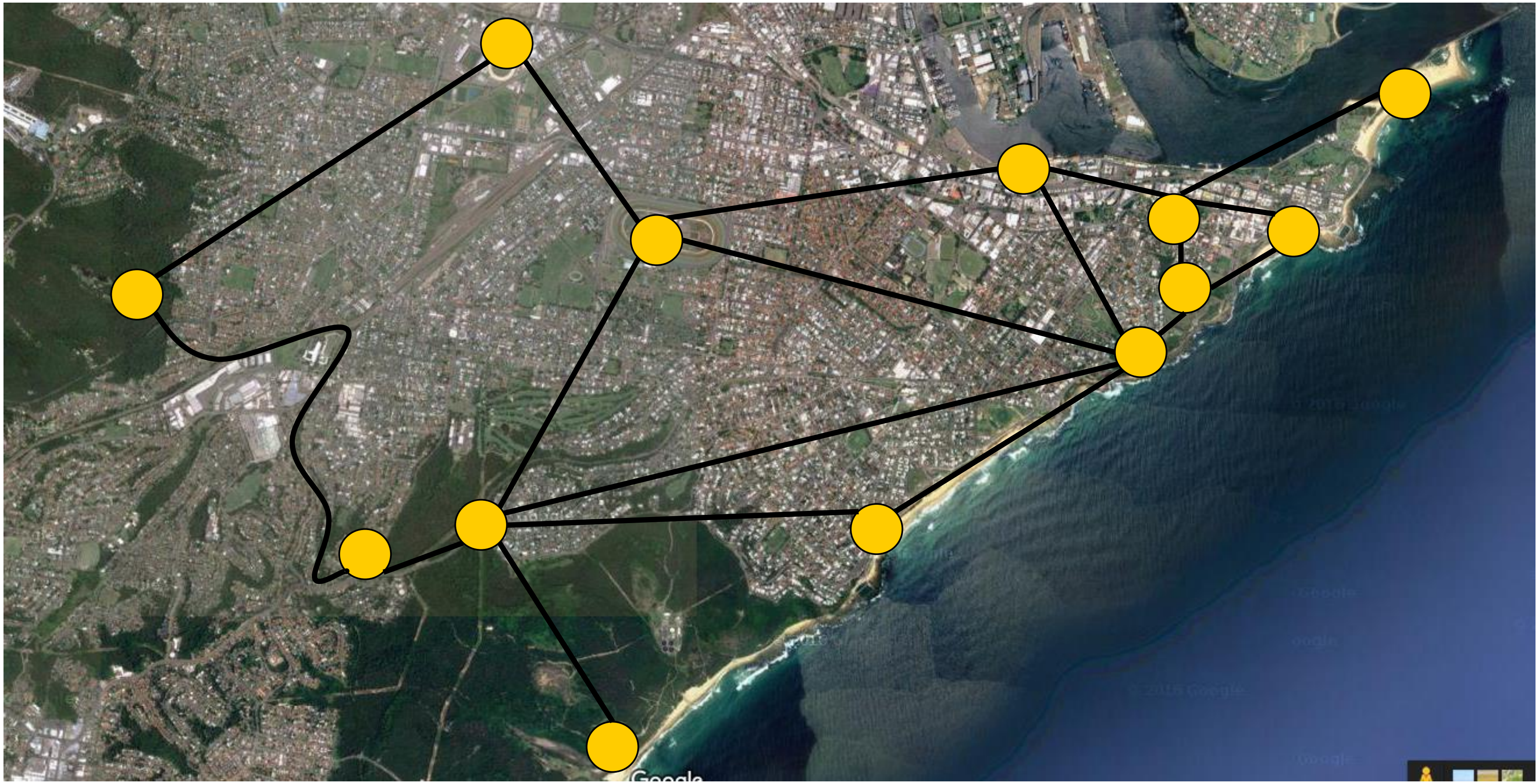
– Prim's Algorithm

Step 1: Select any vertex to be the first vertex of T.

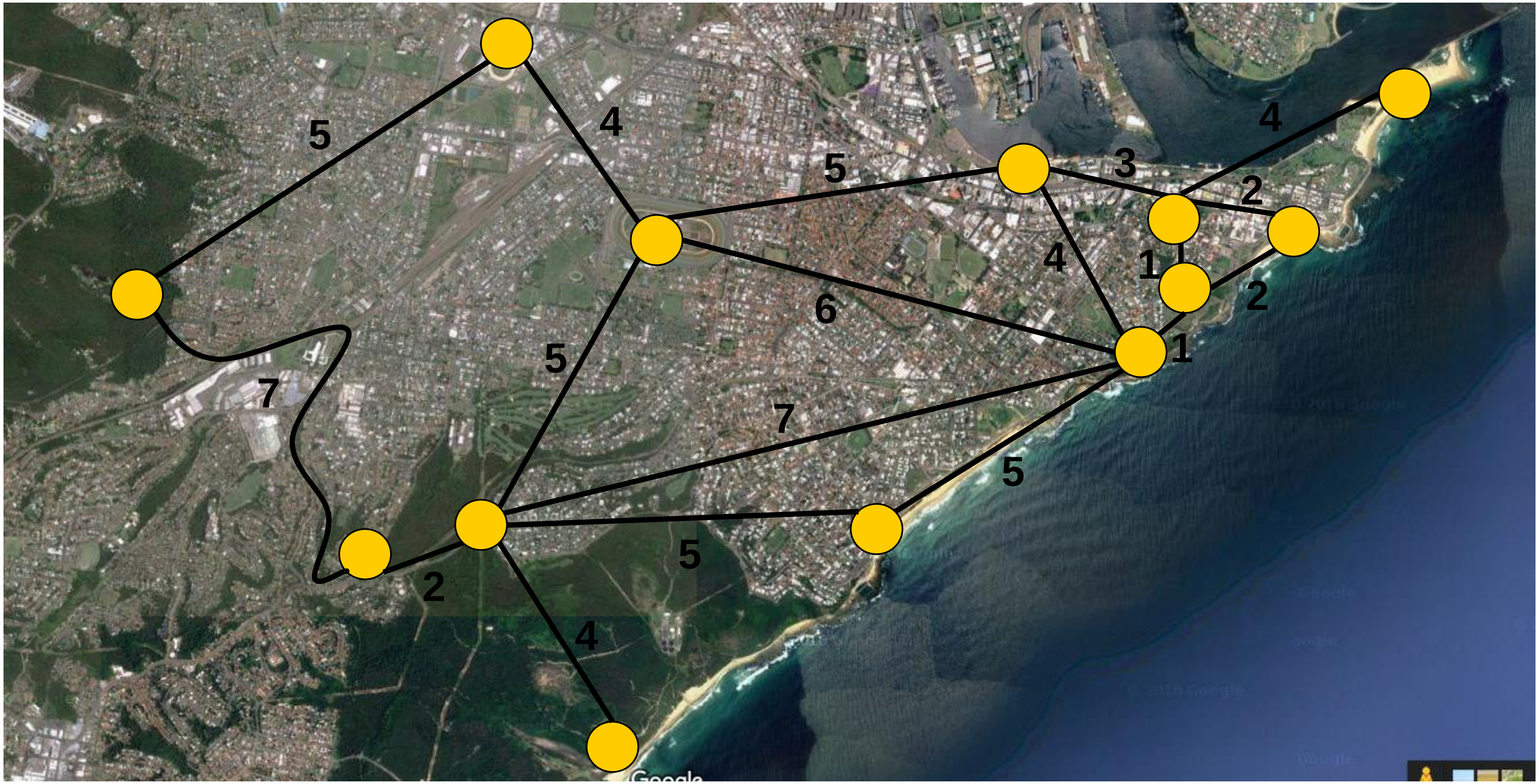
Step 2: Consider the edges which connect vertices in T to vertices outside T. Pick the one with minimum weight. Add this edge and the extra vertex to T. (If there are two or more edges of minimum weight, choose any one of them.)

Step 3: Repeat Step 2 until T contains every vertex of the graph.

Run by hand…

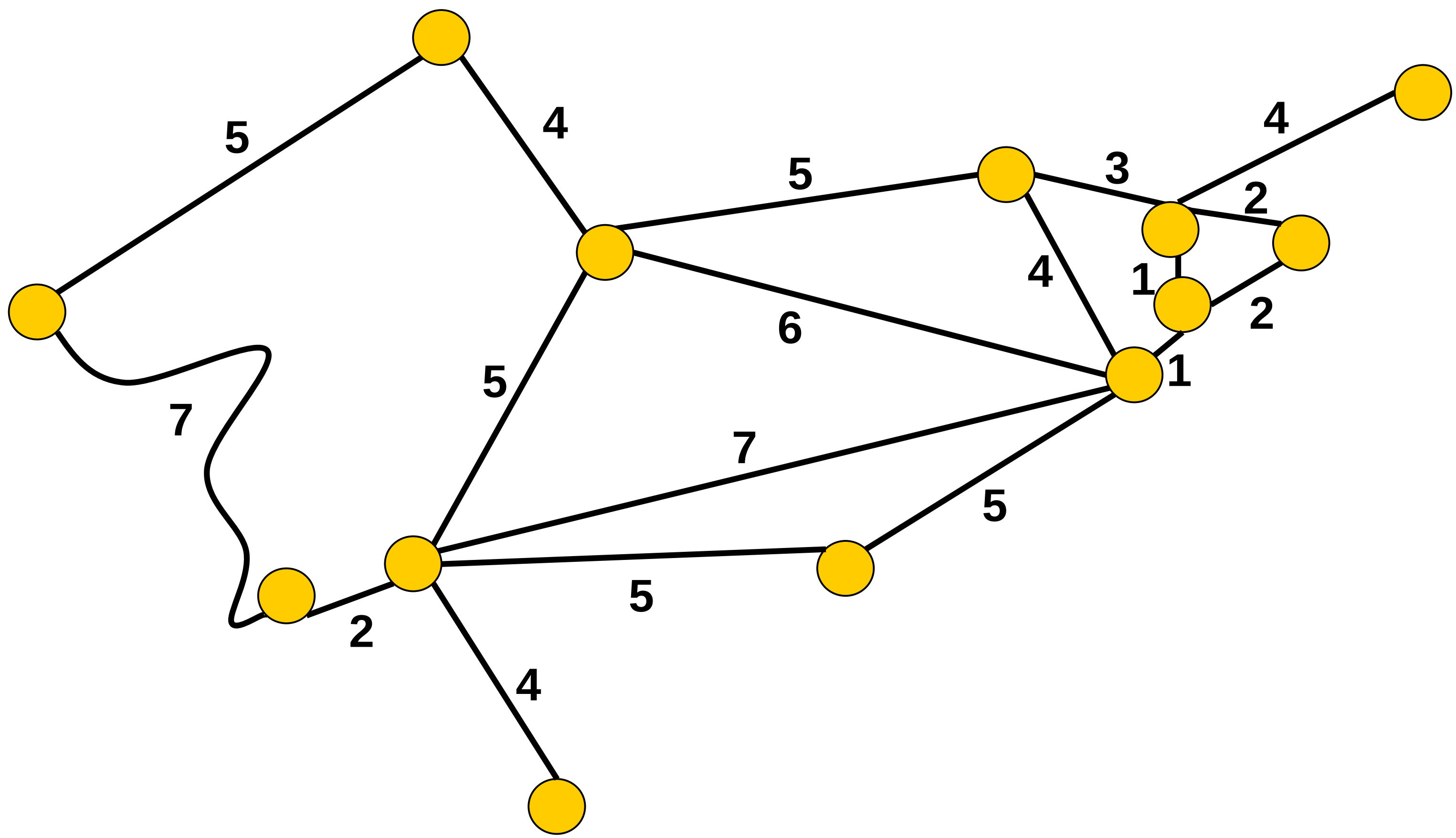


Run by hand…



Run by hand…

(Start at Nobbys Head)



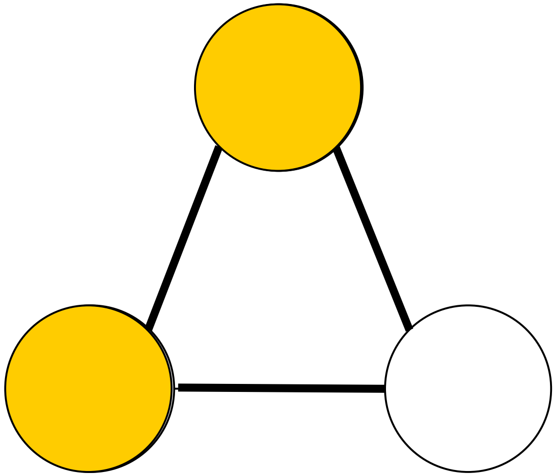
Algorithm…

1. The RTA wants to make sure that they connect all **parks** by at least one super-fast **road**, but they have a very low budget for it. The company that is awarded the contract has to choose the roads that will connect the parks in Newcastle, for the cheapest amount.
2. Council wants to set up a system of security cameras such that for every **road** connecting two **parks** at least one of their endpoints has a camera. Because their budget is also low, they would like the number of cameras to be the minimum possible.
3. The NSW SES wants to establish a system of specialised fire trucks for the **parks** in Newcastle so that each park either has one of these new trucks, or is linked to it by a super-fast **road**.

Algorithm…

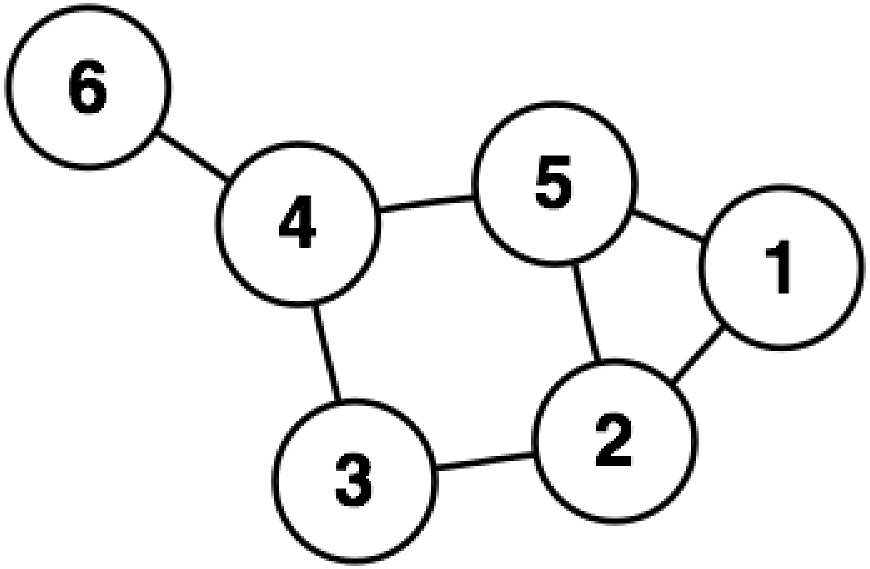
2. Council wants to set up a system of security cameras such that for every **road** connecting two **parks** at least one of their endpoints has a camera. Because their budget is also low, they would like the number of cameras to be the minimum possible.

Vertex Cover



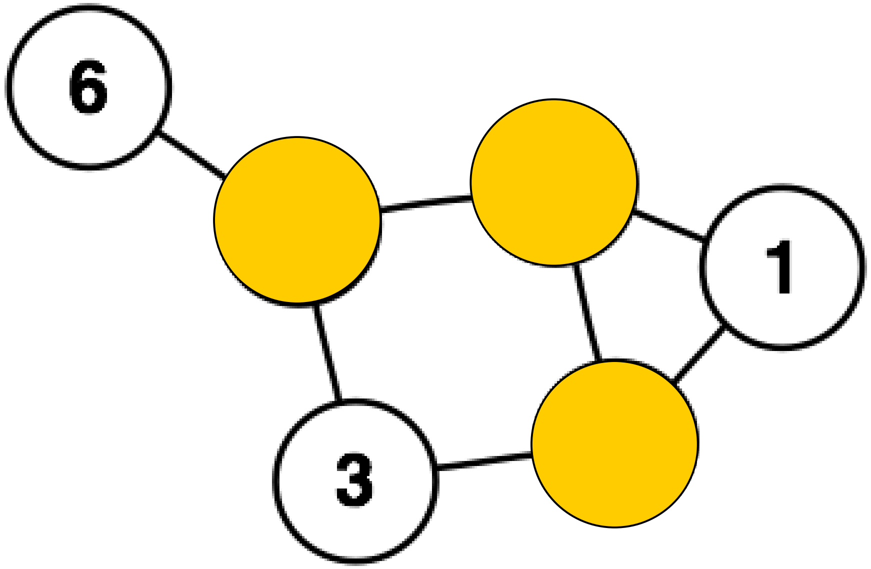
Find the minimum number of cameras to place in the parks so that all roads are "covered"

Vertex Cover



Find the minimum number of cameras to place in the parks so that all roads are "covered"

Vertex Cover



S = {4,5,2}

Can you find any smaller vertex cover?

Is there more than one?

Problem Solving (unplugged)

- Analyse
- Model
- Algorithm
- **Pseudocode**
- Run by hand

Pseudocode for Vertex Cover

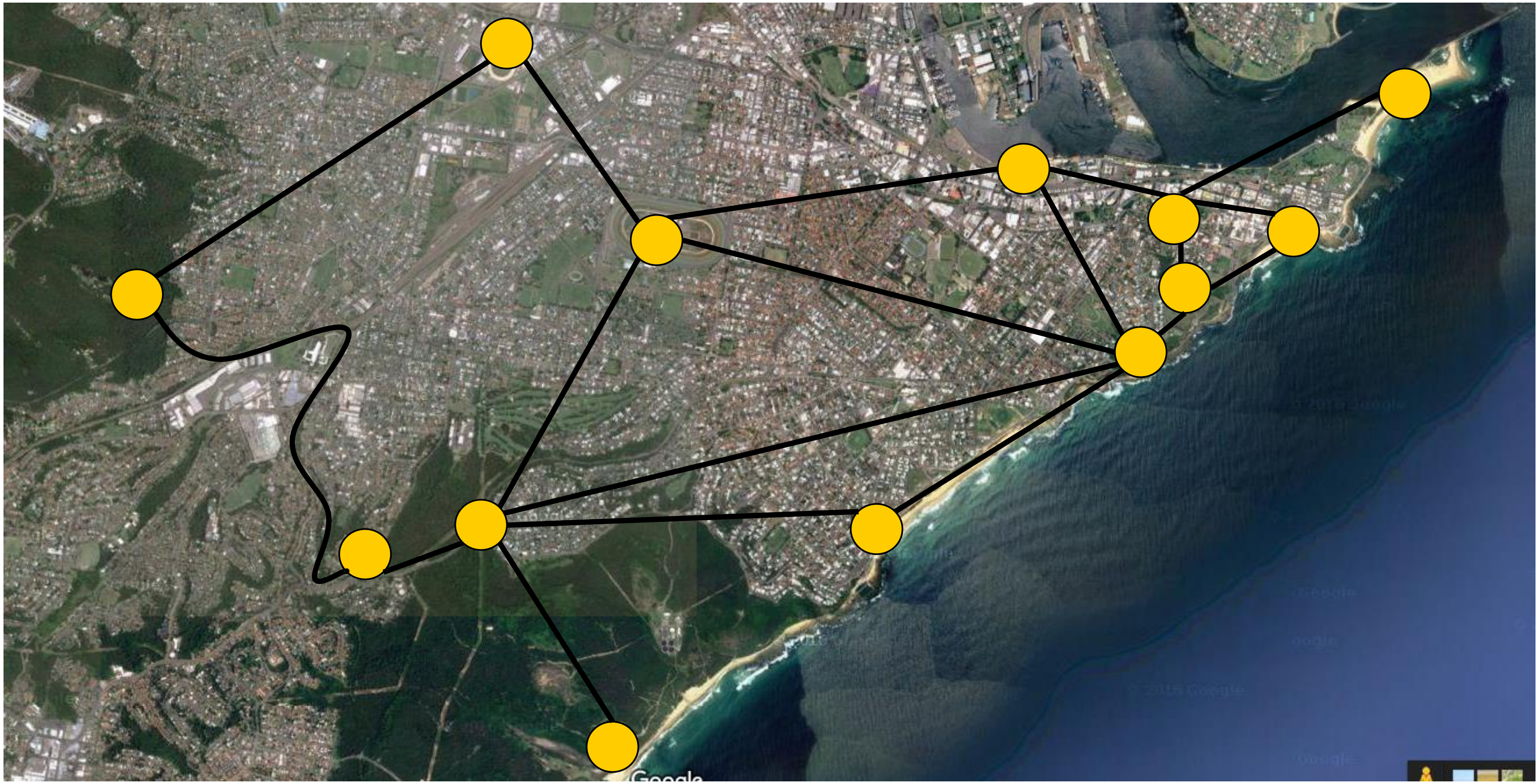
Rule 1: Any pendant vertices will not be in the VC and thus the vertex they are attached to will have to be.

Rule 2: If there is a degree-two vertex with adjacent neighbours, then there is a vertex cover of optimal size that includes both of these neighbours.

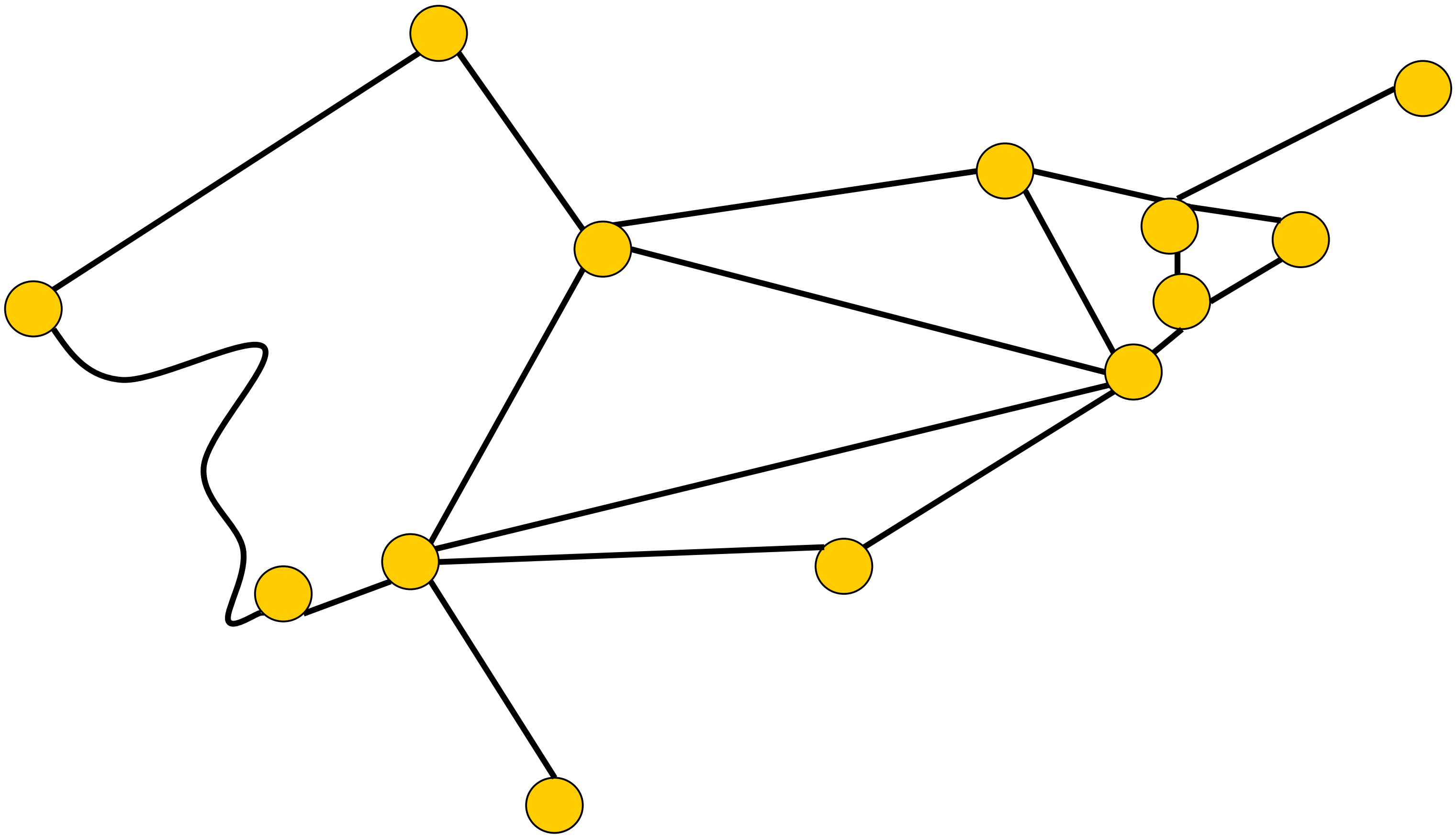
Rule 3: If there is a degree-two vertex, u, whose neighbours, v and w, are non-adjacent, then u can be folded by contracting edges {u, v} and {u, w}… Uhhhh!

There are more…

Run by hand…



Run by hand…



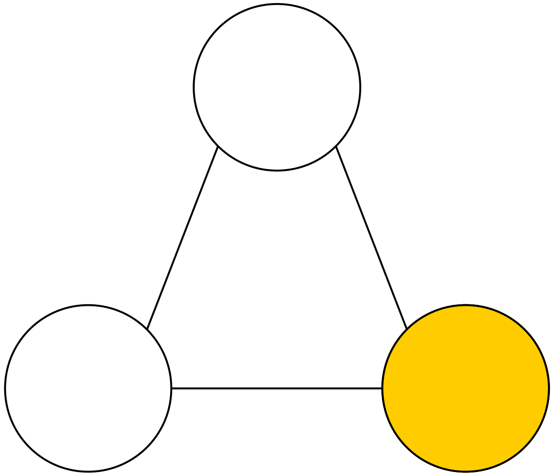
Algorithm…

1. The RTA wants to make sure that they connect all **parks** by at least one super-fast **road**, but they have a very low budget for it. The company that is awarded the contract has to choose the roads that will connect the parks in Newcastle, for the cheapest amount.
2. Council wants to set up a system of security cameras such that for every **road** connecting two **parks** at least one of their endpoints has a camera. Because their budget is also low, they would like the number of cameras to be the minimum possible.
3. The NSW SES wants to establish a system of specialised fire trucks for the **parks** in Newcastle so that each park either has one of these new trucks, or is linked to it by a super-fast **road**.

Algorithm…

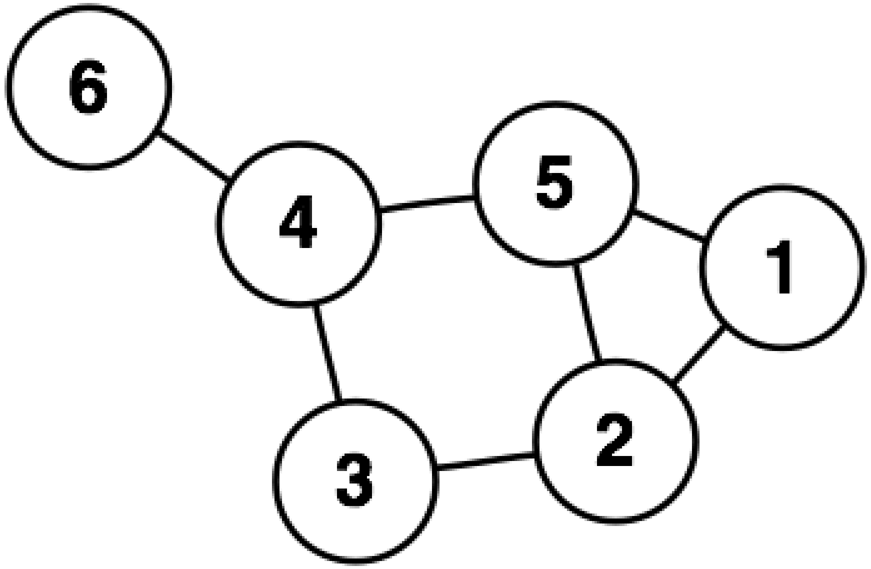
3. The NSW SES wants to establish a system of specialised fire trucks for the **parks** in Newcastle so that each park either has one of these new trucks, or is linked to it by a super-fast **road**.

Dominating set



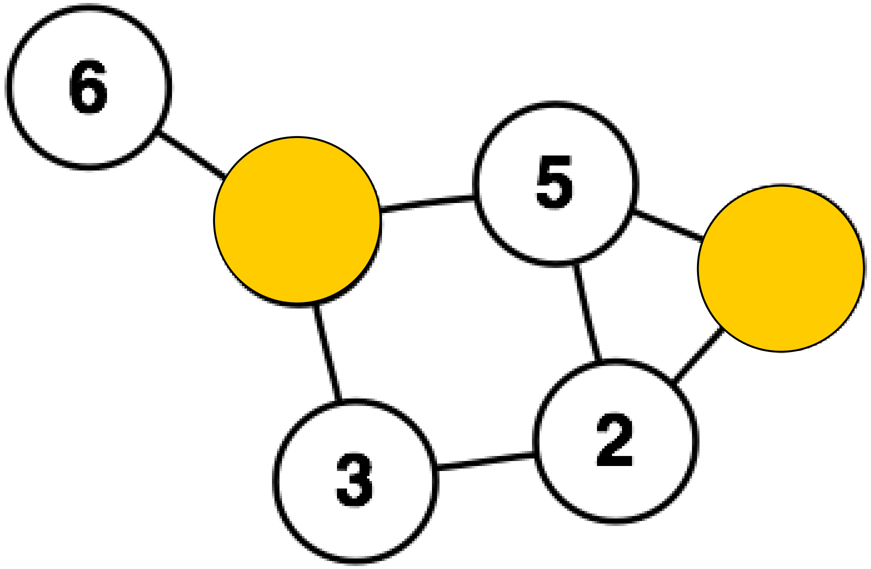
Find the minimum number of trucks to place in the parks so that all other parks are safe

Dominating set



Find the minimum number of trucks to place in the parks so that all other parks are safe

Dominating set



D={4,1}

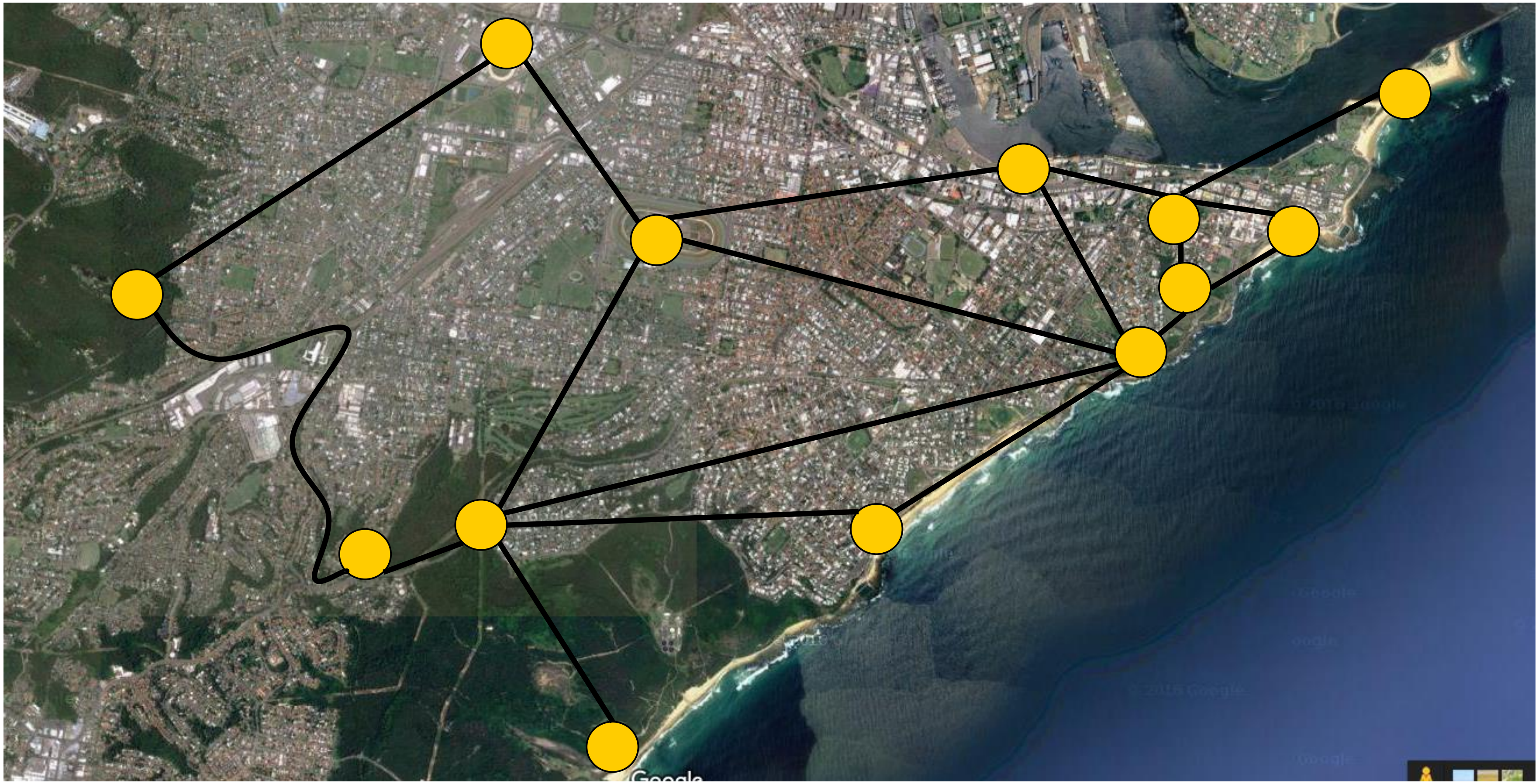
Can you find any smaller dominating set?

Is there more than one?

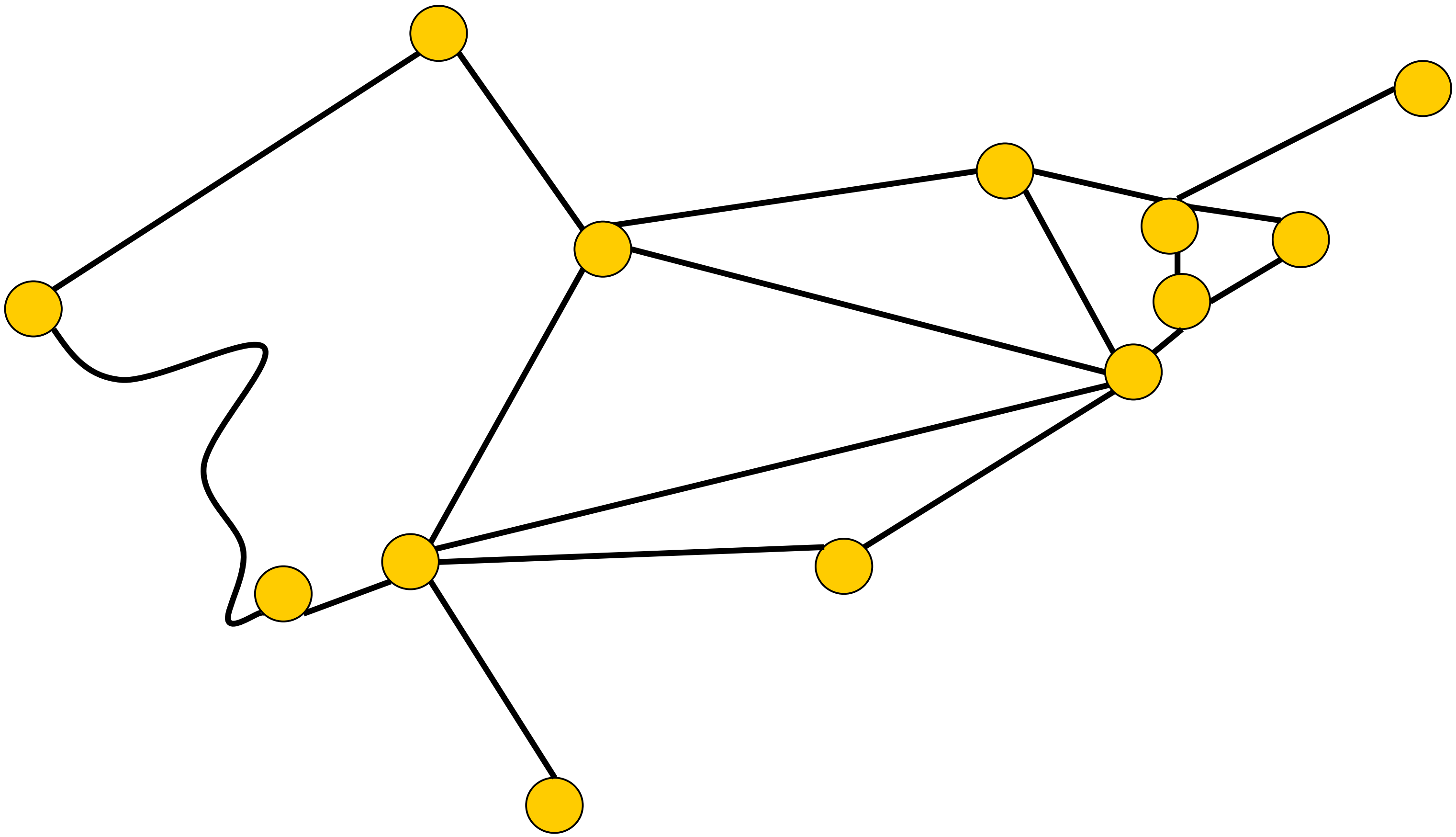
Problem Solving (unplugged)

- Analyse
- Model
- Algorithm
- **Pseudocode?**
- Run by hand!!!

Run by hand…



Run by hand…



Wrap Up

- What problems do you think that are important that could be approached with a Computer Science solution?

More ideas

- Algorithms
  - From the 'simple'…
    - searching and sorting
  - To the complex
    - travelling salesman, graph colouring, Steiner trees

http://csfieldguide.org.nz/en/chapters/algorithms.html

RESOURCES

- **Minimum Spanning Trees**
  - http://csunplugged.org/minimal-spanning-trees
- **Vertex Cover**
  - http://www.siam.org/meetings/alenex04/abstacts/F-Abu-Khzam.pdf
- **Dominating Set**
  - http://csunplugged.org/dominating-sets/