COMP1002
Computational Thinking and Problem Solving

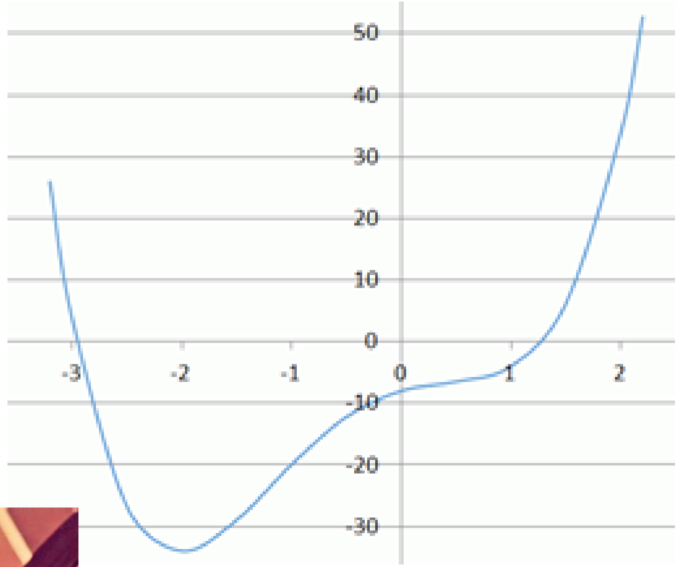
Lecture 8
Problem Solving II

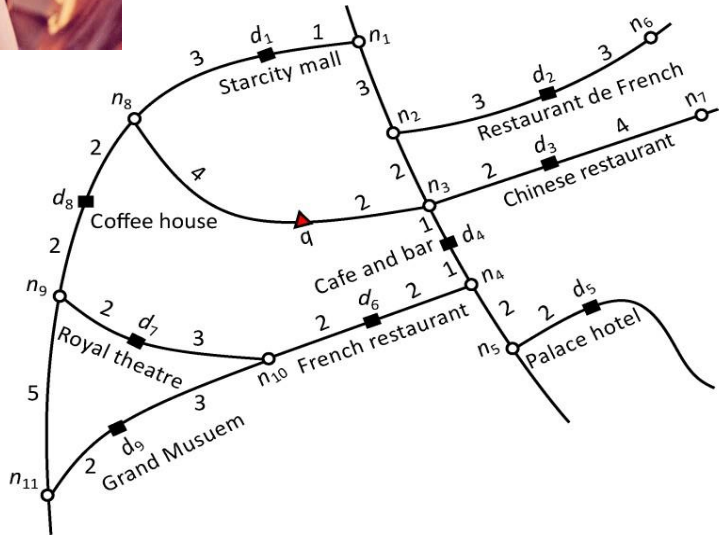
Lecture 8

- Problem Abstraction
  - Graph
- Examples
  - Seven Bridges of Königsberg
  - Map Coloring Problem
  - A Day Change Problem
  - A Vending Machine
  - MCGW Problem

Abstraction

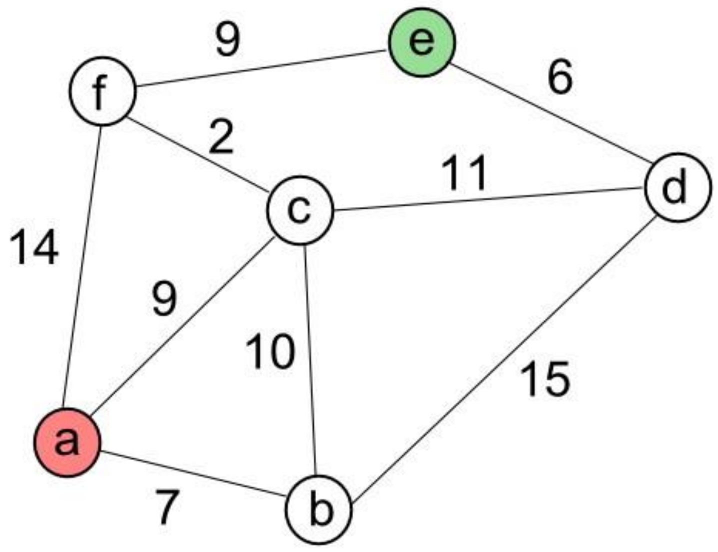
- In computer science, a road network is often modeled as a graph
  - not the *graph* in mathematics
  - also not to be confused with *graphics* in design
  - further not to be confused with *computer graphics*

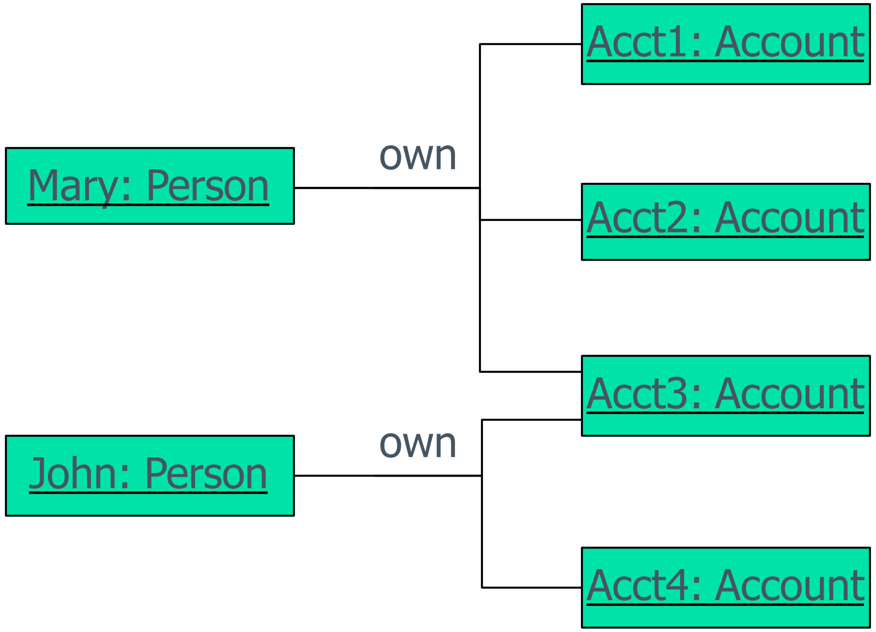




Abstraction

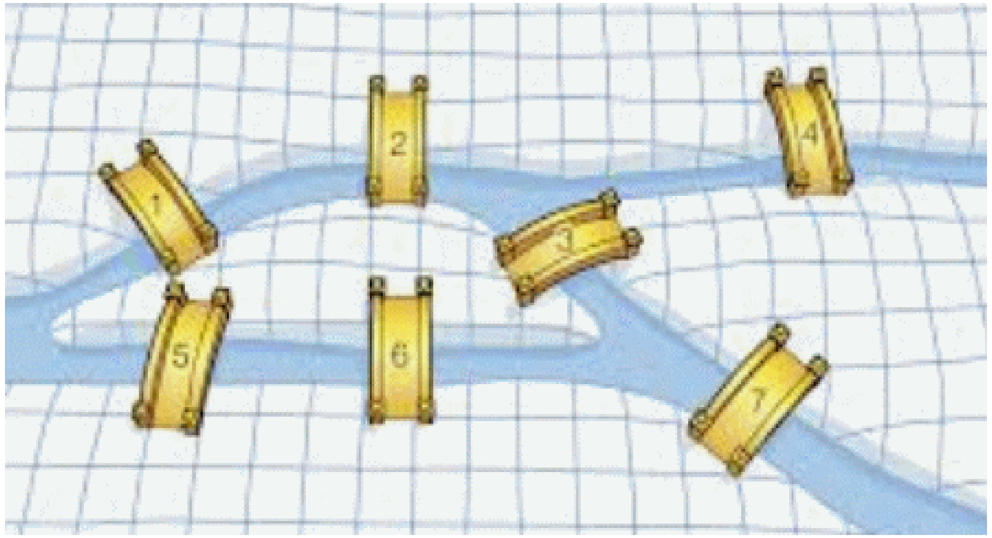
- Why graphs are so common?
  - They naturally represent entities (as nodes) and their relationships (as edges)
    - Database record
    - Object links
    - Social network
    - Any network
    - Any relationship

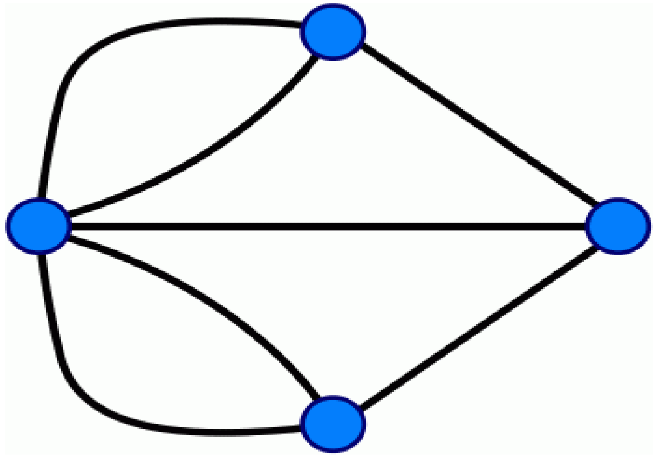




Example

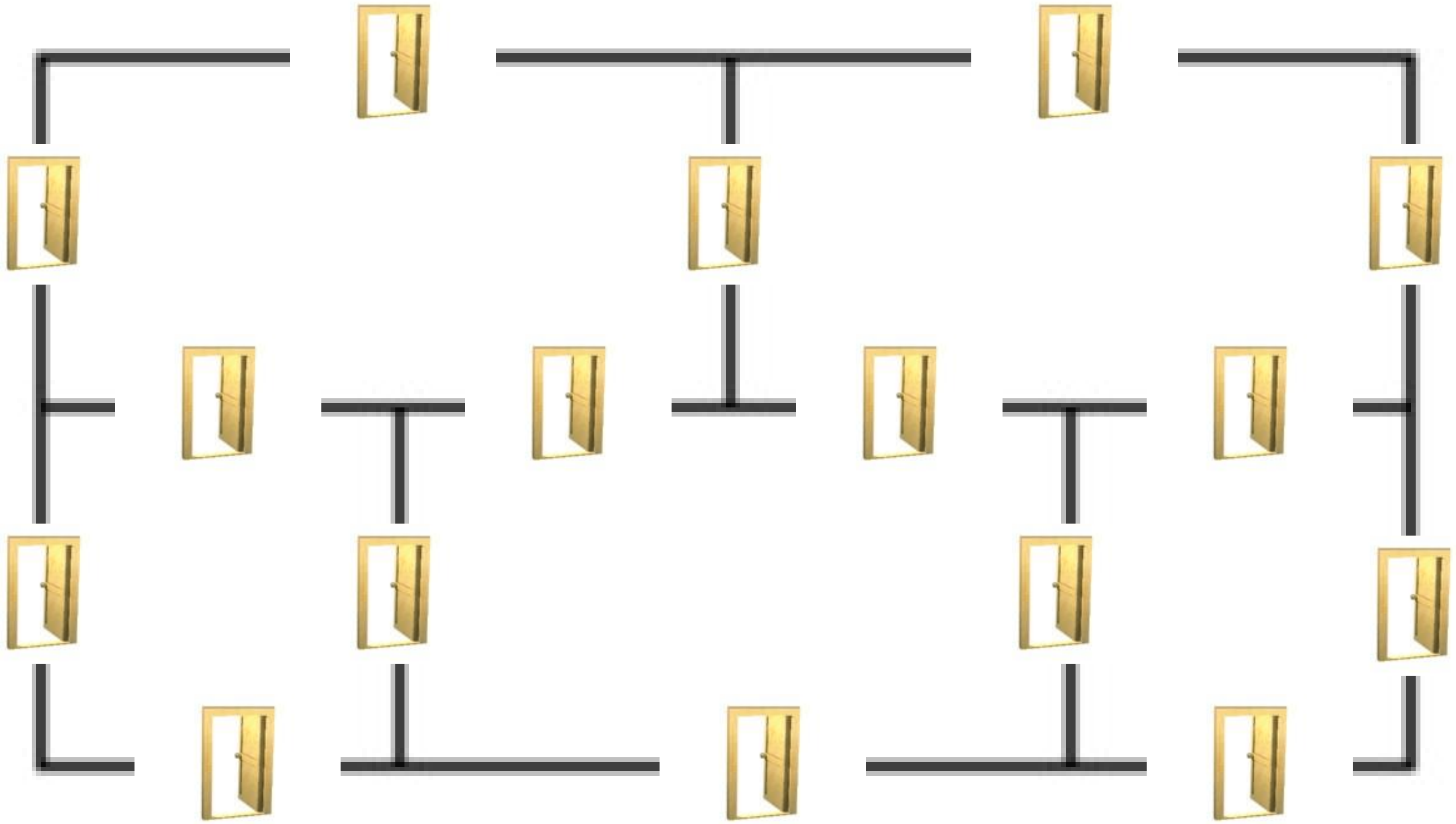
- Seven Bridges of Königsberg
  - The problem is to find a walk through Königsberg (a city in Germany) for a tourist that would cross each of those bridges once and only once
  - Can you do it?
  - How can you prove that there is no solution?
  - We model it as a graph
    - Nodes represent the land block and edges represent the bridges
    - Now the problem becomes finding a path that travels each edge once and only once





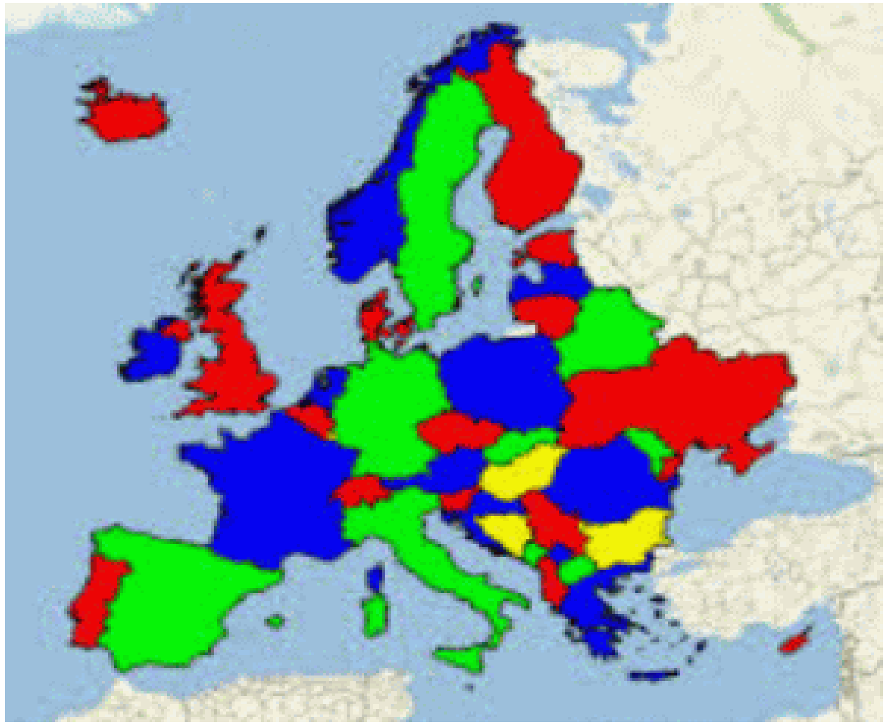
Home Exercise

- Try this



Map Coloring Problem

- Given a map
  - We are to properly color it such that neighboring countries do not share the same color

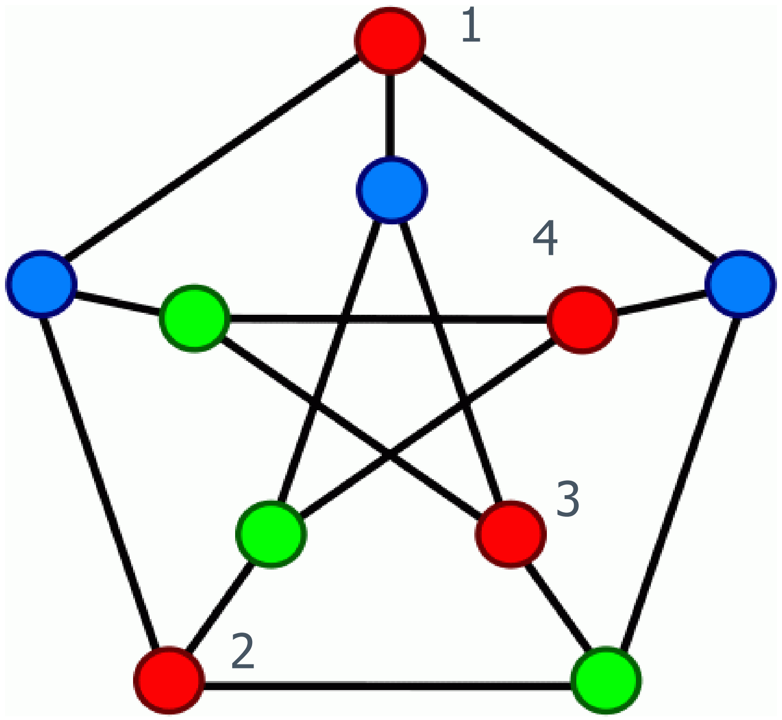


Map Coloring Problem

- Abstraction
  - Convert it into a graph
  - Each country is a node
  - If country A shares border with country B, then there is an edge between A and B
- Observation
  - An edge excludes the use of a same color between the two nodes
  - Such edges are called *conflict edges* as they imply a conflict between connecting nodes

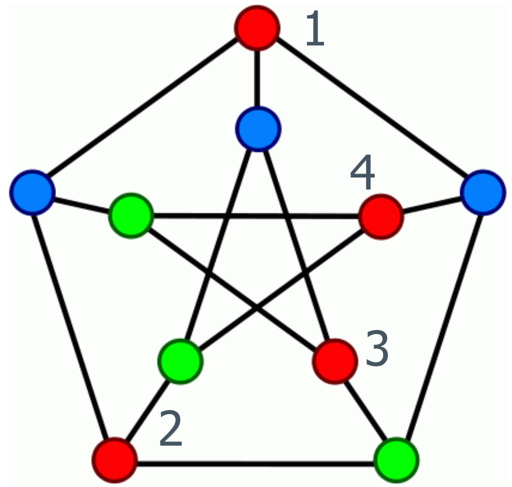
Map Coloring Problem

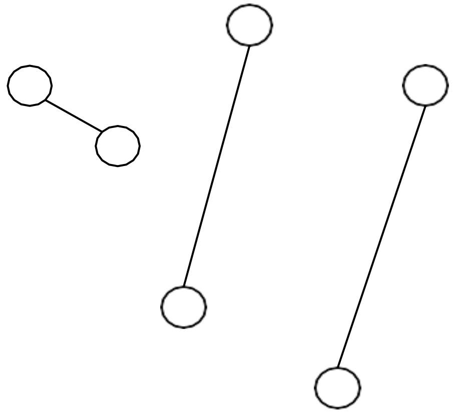
- Solution
  - Since nodes connected with an edge cannot be assigned the same color, remove nodes that have no (conflict) edges among them to be colored first
  - Find an independent set and remove them from the graph, removing also their connecting edges
    - An independent set is a collection of nodes that have no connecting edges within the collection
    - For example, nodes 1, 2, 3, 4 form an independent set



Map Coloring Problem

- Solution (cont')
  - Repeat the procedure for the remaining nodes, each set receiving a color, until nothing is left
  - Example
    - Starting from top red color 1, try other possible nodes to be colored red, e.g., 1, 2, 3, 4
    - We remove red nodes and edges linked to them
    - We could color 3 as blue and remaining as green





A Day Change Problem

› How could we model a software button to adjust the day for today (Monday)

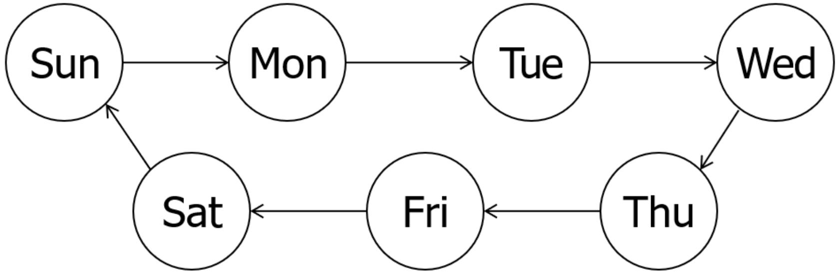
Sunday ⇒ Monday ⇒ Tuesday ⇒ Wednesday ⇒ Thursday ⇒ Friday ⇒ Saturday ⇒ Sunday …

– Each day can be associated with different actions, e.g., subjects to attend, assignments to work on, etc.

```
If button pressed then
    if day = Sunday then day = Monday; take Monday actions
    else if day = Monday then day = Tuesday; take Tuesday actions
    else if day = Tuesday then day = Wednesday; take Wednesday actions
    …
    else if day = Saturday then day = Sunday; take Sunday actions
```

A Day Change Problem

- Note that the changes occur in a *cycle*



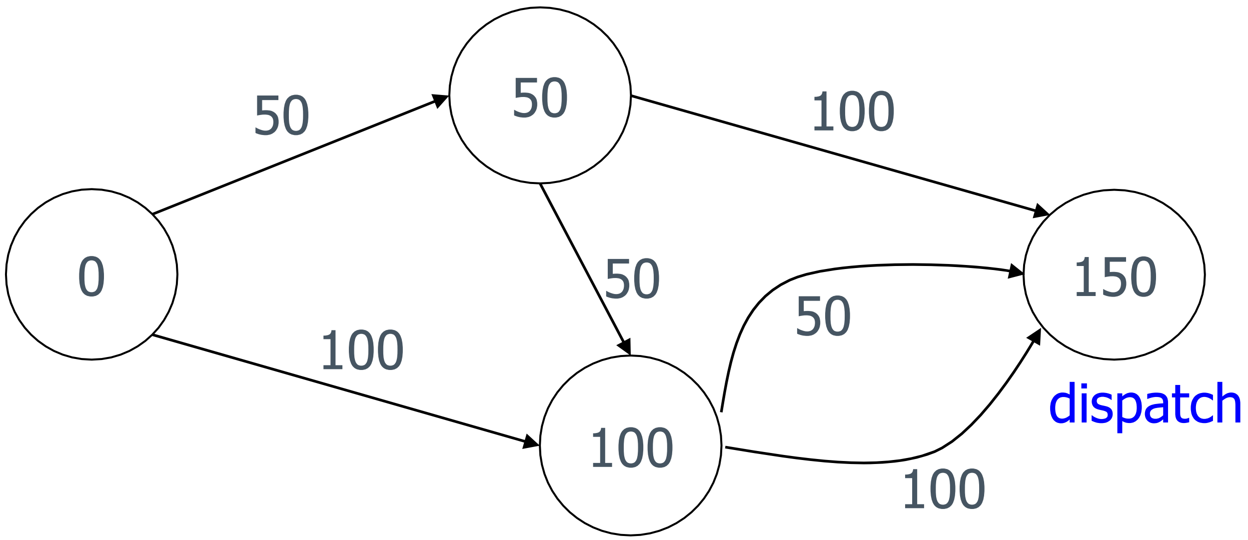
- This can be modeled as a graph
  - Nodes are the days
  - Edges (with direction) represent the change of days
- We call it a *state diagram* or *state transition diagram*
  - Nodes are states
  - Edges are transitions

A Vending Machine

- Consider a vending machine selling candy costing \$1.50
  - It accepts only \$1 and 50c coins
- How can we model this vending machine?
- We model the state of the machine by the amount of money paid
  - There are 4 possible states 0, 50, 100, 150
  - It is clear that 0 is the beginning state of the machine and 150 is the final state of the machine

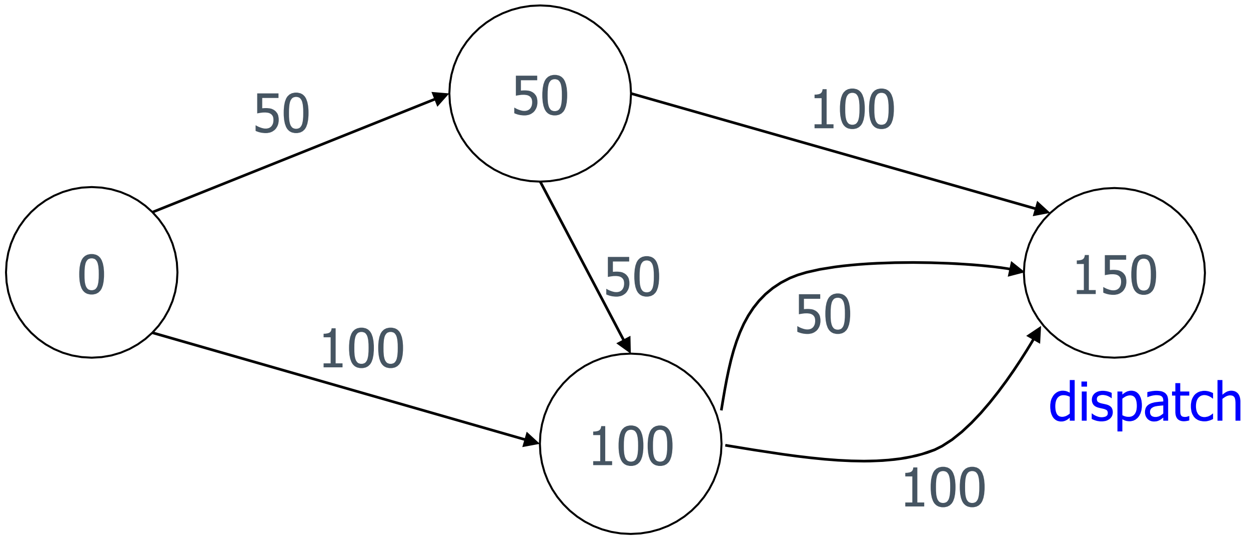
A Vending Machine

- If we pay 50c from state 0, it will take us to state 50
- If we pay $1 from state 0, it will take us to state 100
- We continue with from state 50, and from state 100
- Note that even if you pay $1 from state 100, it just brings you to state 150 (and dispatch the candy)



Home Exercise

- How could you add transitions to model the machine capable of accepting a $2 coin?
- How could you model coin changing for this vending machine?



MCGW Problem

› Man, Cabbage, Goat, Wolf problem

› Problem
  - Bring the cabbage, goat and wolf from the East (right) side of the river to the West (left) side

› Constraints
  1. The man can bring only one of the cabbage, goat or wolf at any time
  2. The cabbage cannot stay with the goat alone or it will eat the cabbage
  3. The goat cannot stay with the wolf alone or it will eat the goat

MCGW Problem

- How do you solve this problem?
  - In an ad hoc manner?
- A more systematic way
  - Start with a picture showing all on East side of the river
  - Draw successive pictures to show the changing situations
  - A solution is found with a picture shows that all are on the West side of the river
  - Trace back for the steps leading to the target situation
- A picture shows a state of the system and a link between two pictures shows a transition
  - The problem can be modeled as a state transition diagram, or a graph
  - The goal is to start from a certain start state (all on east) and go to a certain target state (all on west)

MCGW Problem

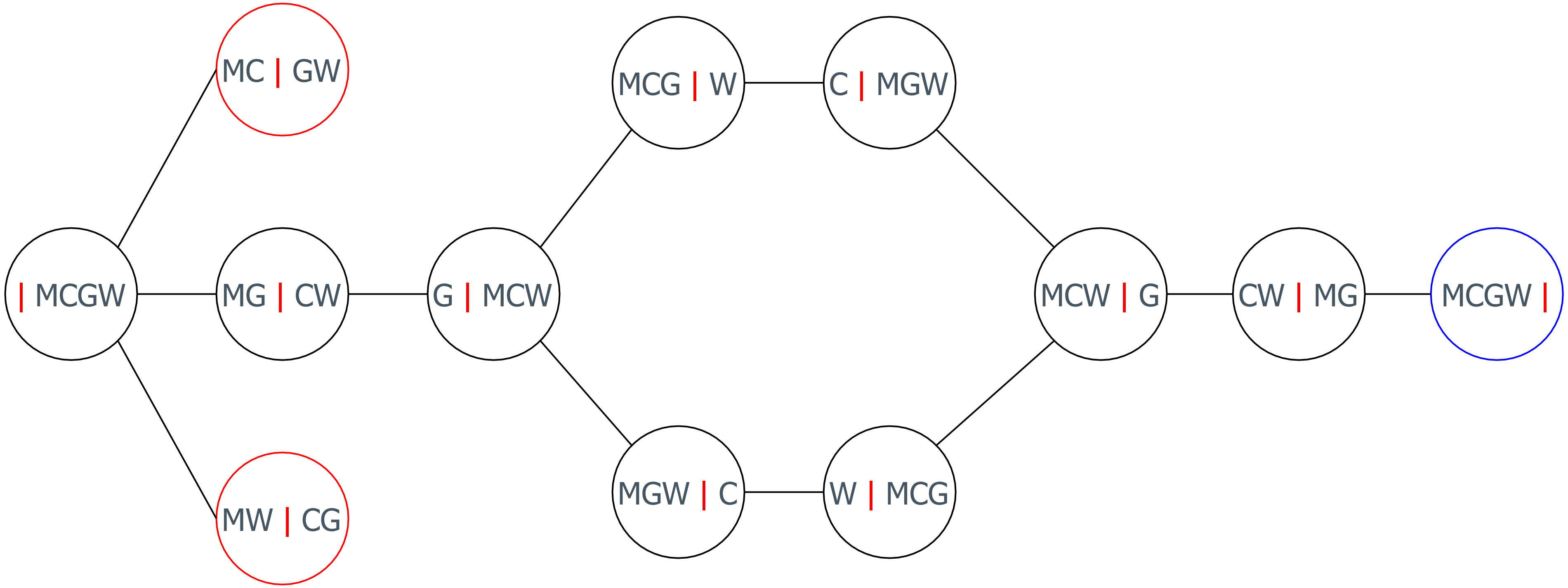
- After understanding the problem, next is abstraction
- Recall that abstraction is a representation of a problem with just enough important details
- Let us denote the river by “|” and MCGW for man, cabbage, goat and wolf respectively
- At the beginning, we have “| MCGW” and our goal is “MCGW |”
- There are four possible moves from the beginning:
  | MCGW -> M | CGW
  | MCGW -> MC | GW
  | MCGW -> MG | CW
  | MCGW -> MW | CG
- Any problem with the moves above?

MCGW Problem

- After making the first “good” move, we have 2 possible moves
  - | MCGW -> MG | CW -> | MCGW
  - | MCGW -> MG | CW -> G | MCW
- Now, we have 3 possible moves from the current situation
  - G | MCW -> MG | CW
  - G | MCW -> MCG | W
  - G | MCW -> MGW | C

MCGW Problem

- Putting them together in a graph
  - 7 steps, 2 possible answers



MCGW Problem

Why tuple is preferred, not list in Python?

- Another possible model
- We assign East (E) or West (W) to indicate the current location of man, cabbage, goat and wolf
  - Man: E/W
  - Cabbage: E/W
  - Goat: E/W
  - Wolf: E/W
- A state is a collection of 4-tuples (E/W, E/W, E/W, E/W) for [man, cabbage, goat, wolf]
- The problem is thus to bring from (E, E, E, E) state to (W, W, W, W) state without losing the cabbage and goat

MCGW Problem

- Starting from (E, E, E, E) for [man, cabbage, goat, wolf], try the three possible moves

  (E, E, E, E) $\Rightarrow$ (W, W, E, E)
  (E, E, E, E) $\Rightarrow$ (W, E, W, E)
  (E, E, E, E) $\Rightarrow$ (W, E, E, W)
- On the return trip by man

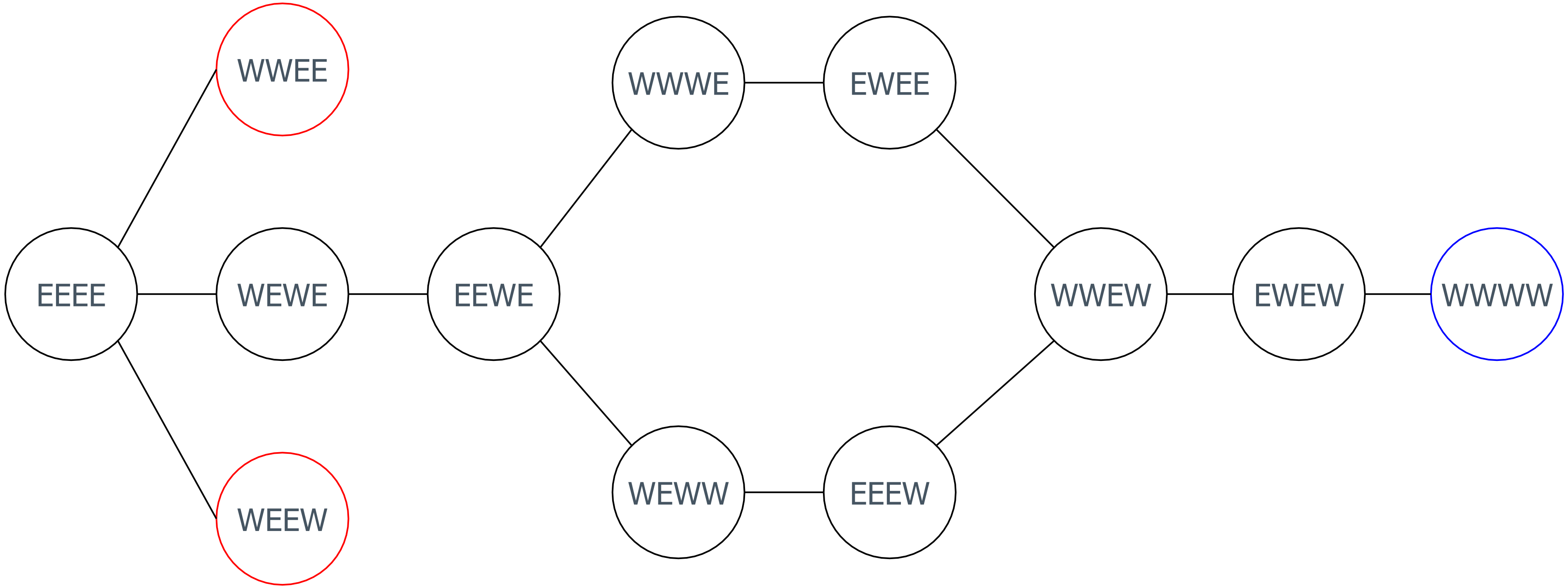
  (W, E, W, E) $\Rightarrow$ (E, E, W, E)
- Now we are back to the East side, repeat the steps

  (E, E, W, E) $\Rightarrow$ (W, W, W, E)
  (E, E, W, E) $\Rightarrow$ (W, E, W, W)
- Find the ones that meet the solution requirements
- Can you draw pictures to help understanding?

MCGW Problem

[man, cabbage, goat, wolf]

- Another graph
  - Still 7 steps
  - 2 possible answers



MCGW Problem

- What are the differences between the two models?
  - They are the same with the same number of nodes and links
  - The major difference is the representation of information inside each node
  - The first one is easier to understand for human
  - The second one is easier to represent inside a computer
  - Each node can be represented as a 4-tuple
- If you are to solve this problem using a computer, it is better to adopt the second model
- A solution is found when there is a path leading from the start node to the end node
- The path tells the steps needed

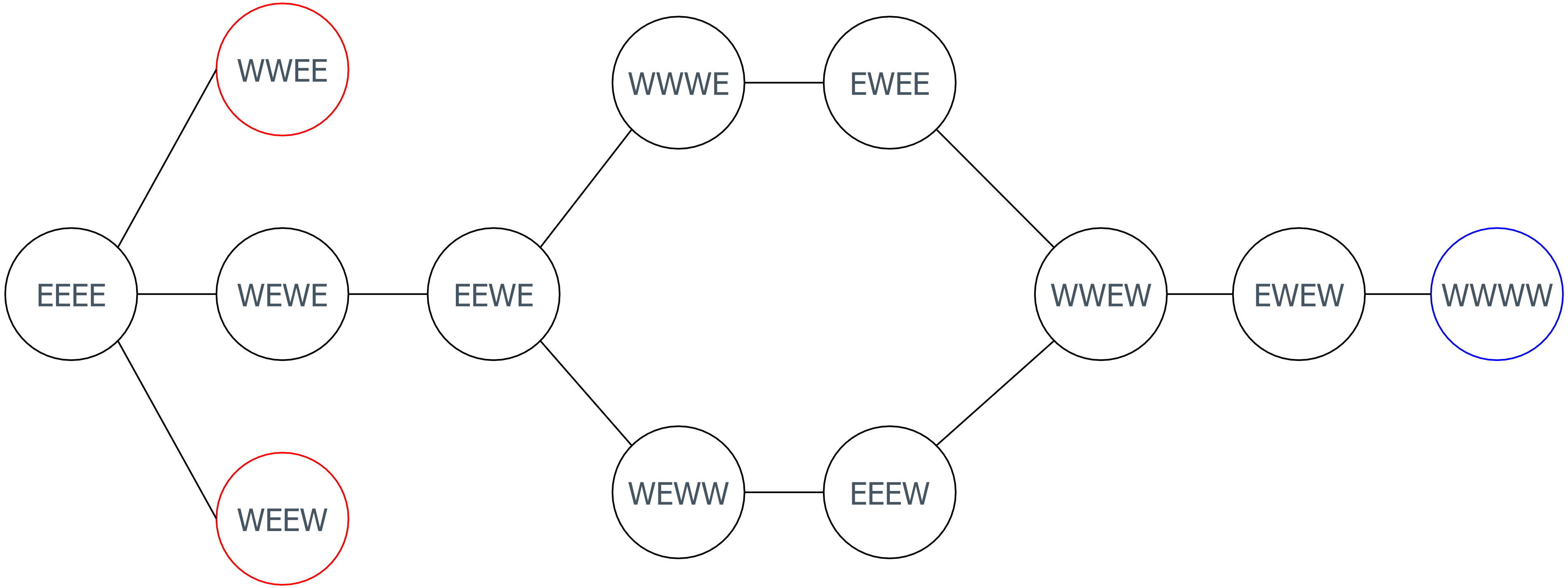
[man, cabbage, goat, wolf]

MCGW Problem

› Is it really true that there are 2 possible answers?

EEEE ⇒ WEWE ⇒ EEWE ⇒ WWWE ⇒ EWEE ⇒ WWEW ⇒ EWEW ⇒ WWEW ⇒ EWEW ⇒ WWWW

EEEE ⇒ WEWE ⇒ EEWE ⇒ WWWE ⇒ EWEE ⇒ WWEW ⇒ EEEW ⇒ WEWW ⇒ EEWE ⇒ WWWE ⇒ EWEE ⇒ WWEW ⇒EWEW ⇒ WWWW



MCGW Problem

[man, cabbage, goat, wolf]

- Some related concepts
  - The best solution is one that contains smallest number of steps: only 2 best solutions with 7 steps
  - An acceptable solution is one that contains no cycle (e.g., EWEW $\Rightarrow$ WWEW $\Rightarrow$ EWEW is considered a cycle)
  - A legal state is a state that is correct
  - An illegal state is a state that is not correct or forbidden
    - In MCGW problem, an illegal state is a state that the wolf would eat the goat, or the goat would eat the cabbage (in the absence of the man)
      - Wolf eating goat: (E,E/W,W,W), (W,E/W,E,E)
      - Goat eating cabbage: (E,W,W,E/W), (W,E,E,E/W)
    - So, how many illegal states are there in this problem?

MCGW Challenge

- Consider the lion and wildebeest problem
  - 3 lions and 3 wildebeests need to cross the river
  - Boat can carry at most two animals
  - All can row the boat
  - If number of lions is more than number of wildebeests on any side of the river, the lions will eat the wildebeests
  - Can you solve this problem?
    - What information should be stored in a node?
    - How many nodes are there in the graph?

MCGW Challenge

- Consider the three-couple problem
  - 3 husbands and 3 wives need to cross the river
  - Boat can hold at most two persons
  - All can row the boat
  - If a wife is not with her husband, other husbands there will do bad thing
- Can you solve this problem?
  - What information should be stored in a node?
  - How many nodes are there in the graph?

Summary

› Problem Abstraction
  – Graph
› Examples
  – Seven Bridges of Königsberg
  – Map Coloring Problem
  – A Day Change Problem
  – A Vending Machine
  – MCGW Problem