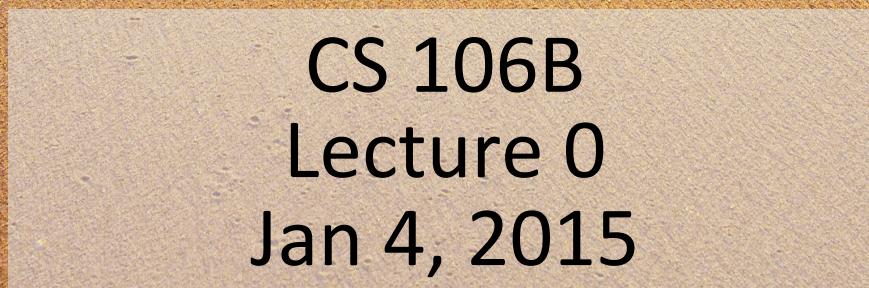




# Hello CS106B

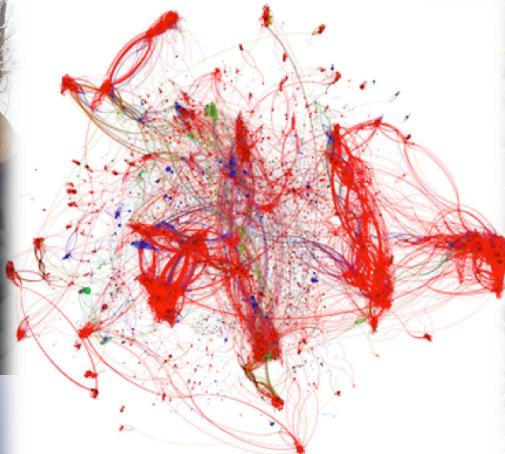
WELCOME



CS 106B  
Lecture 0  
Jan 4, 2015

# Hi, I'm Chris

Chris Piech (piech@cs.stanford.edu)



# Course Staff



Instructor: Chris Piech



Head TA: Megan Faulk

# Who are You?

# Who are You?

- African Studies
- Applied Physics
- Bioengineering
- Biology
- Business Administration
- Chemical Engineering
- Chemistry
- Classics
- Civil and Environmental Engineering
- Computational and Mathematical Engineering
- Computer Science
- Creative Writing
- East Asian Studies
- Economics
- Education
- Electrical Engineering
- Energy Resource Engineering
- English
- Financial Mathematics
- Film and Media Studies
- French
- History
- International Relations
- Japanese
- Law
- Materials Science and Engineering
- Mathematical and Computational Sciences
- Mathematics
- Mechanical Engineering
- Medicine
- Management Science and Engineering
- Modern Language
- Music
- Neuroscience
- Physics
- Political Science
- Psychology
- Science, Technology, and Society
- Statistics
- Symbolic Systems
- Undeclared!

Logistics

Course Overview

First Program

Excited?

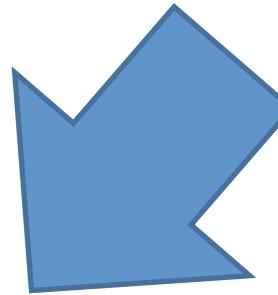
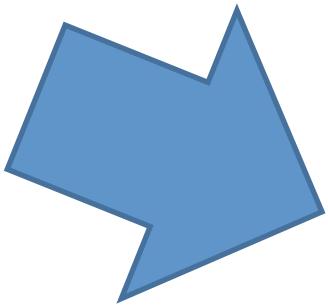
# Logistics

## Course Overview

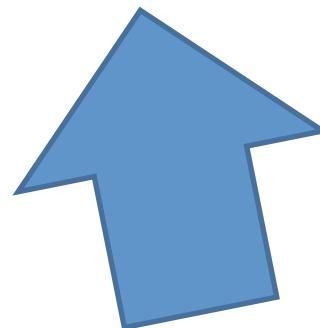
## First Program

## Excited?

# Essential Information



[cs106b.stanford.edu](http://cs106b.stanford.edu)



# Candy Policy



# Assignments in CS106B

- Due at 5:00P.M.
- Three free “late days”
- Extensions approved by Megan or Chris.
- Graded by your section leader
- Interactive, one-on-one grading session.
- Graded on Style and Functionality.

# Grading Scale

Functionality and style grades for the assignments use the following scale:

- ++** A submission so good it “makes you weep.”
- +** Exceeds requirements.
- ✓ +** Satisfies all requirements of the assignment.
- ✓** Meets most requirements, but with some problems.
- ✓ -** Has more serious problems.
- Is even worse than that.
- Better than nothing.

# Sections

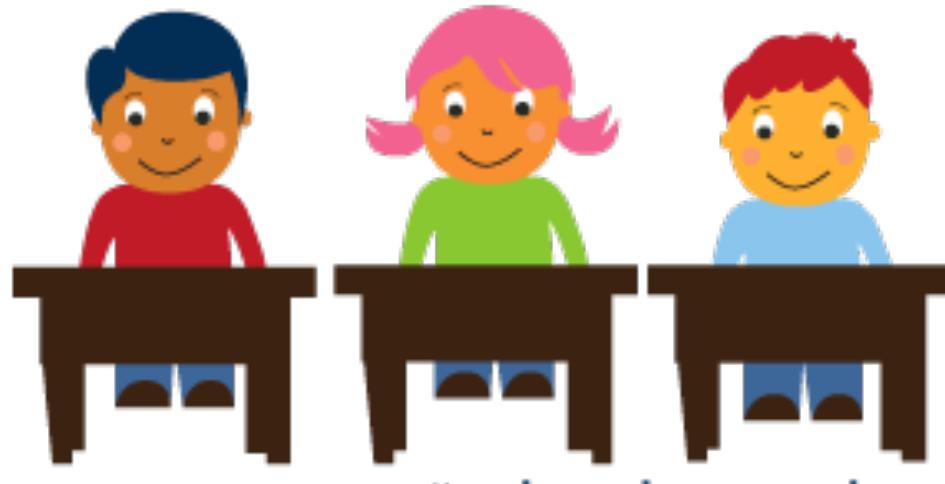
- Weekly 50-min section led by awesome section leaders (the backbone of the class!)
- Signups begin on Thursday at 5:00pm and close Sunday at 5:00pm



# Staff Contact

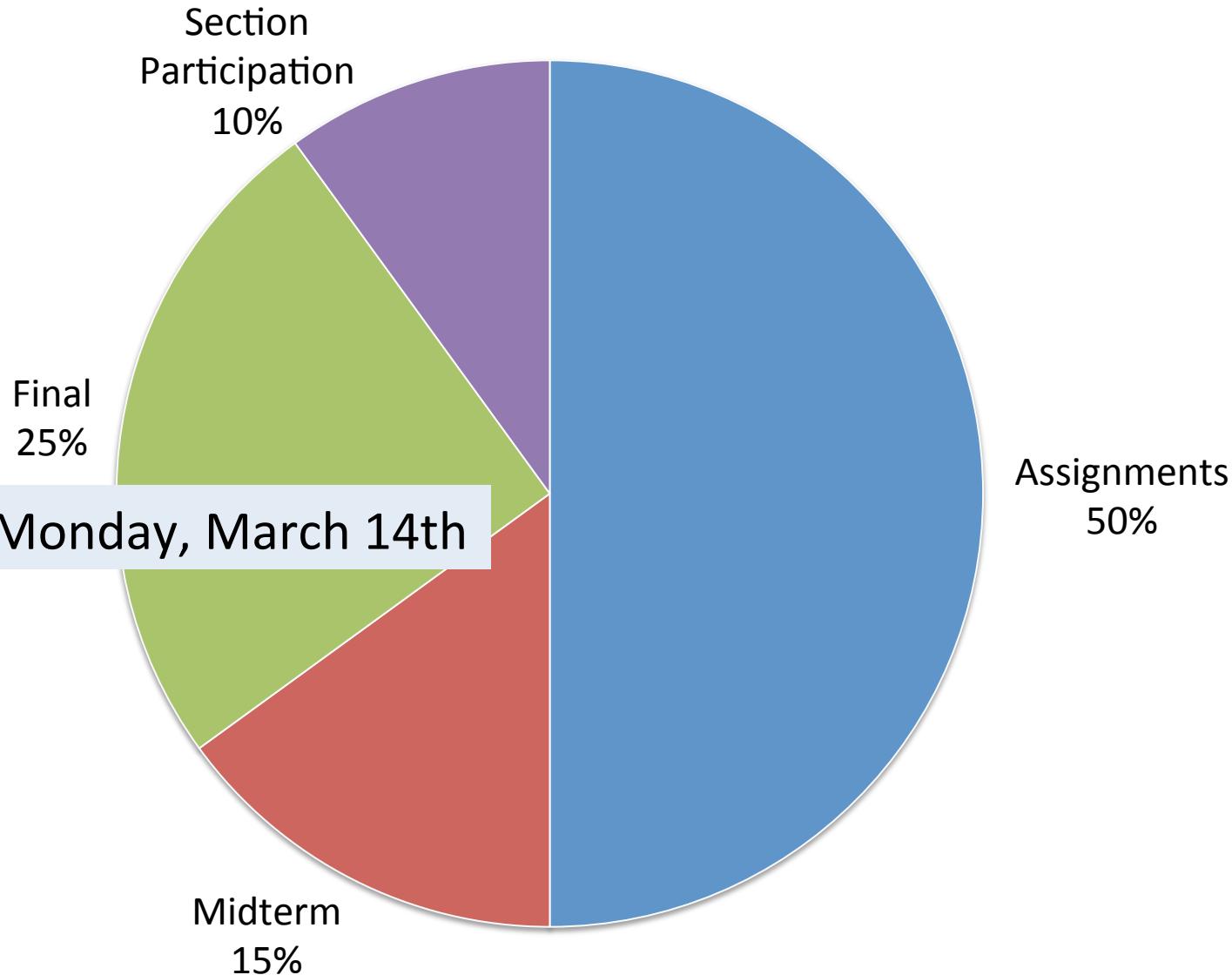
- Post to Piazza
- Contact your Section Leader
- Go to the LaIR or Office Hours
- Email [cs106b-win1516-staff@lists.stanford.edu](mailto:cs106b-win1516-staff@lists.stanford.edu)
- Call us.

# Attendance



#schooleveryday

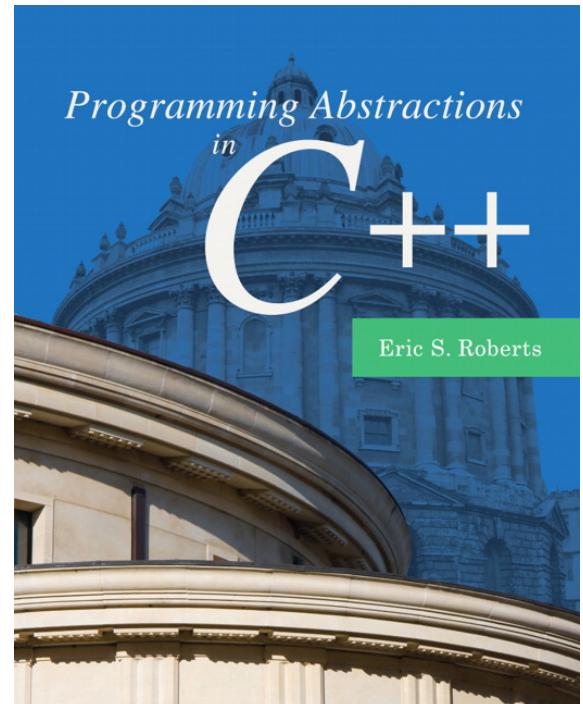
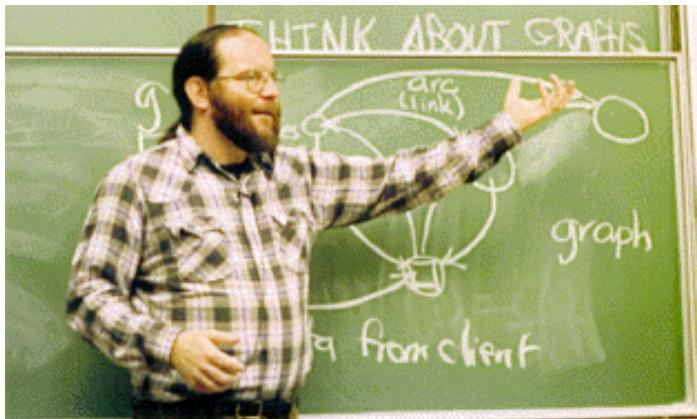
# Grade Breakdown



# Textbook

*Programming Abstractions in C++, by Eric Roberts*

- written here at Stanford
- tailored to this course
- a valuable reference
- usable on open-book exams



# Honor Code

- Rule 1: You must not share actual program code with other students.
- Rule 2: You must not look at solutions posted on the web or from other years.
- Rule 3: If you realize that you have made a mistake, you may retract your submission to any assignment at any time, no questions asked.

# How Many Units?

```
int numUnits(bool isGrad, bool wantsFewerUnits) {  
}  
}
```

# How Many Units?

```
int numUnits(bool isGrad, bool wantsFewerUnits) {  
    if (!isGrad) return 5;  
  
}  
}
```

# How Many Units?

```
int numUnits(bool isGrad, bool wantsFewerUnits) {  
    if (!isGrad) return 5;  
    if (!wantsFewerUnits) return 5;  
}  
}
```

# How Many Units?

```
int numUnits(bool isGrad, bool wantsFewerUnits) {  
    if (!isGrad) return 5;  
    if (!wantsFewerUnits) return 5;  
  
    if (reallyBusy()) {  
        return 3;  
    }  
}
```

# How Many Units?

```
int numUnits(bool isGrad, bool wantsFewerUnits) {  
    if (!isGrad) return 5;  
    if (!wantsFewerUnits) return 5;  
  
    if (reallyBusy()) {  
        return 3;  
    } else {  
        return 4;  
    }  
}
```

# Getting Help

- LaIR Hours!
  - Sunday – Thursday, 6PM – Midnight
  - Starts next week.
- Chris' Office Hours in Gates 193
  - Monday 1:30pm – 2:30pm
  - Wednesday 2:30PM – 4:00PM
- Megan's Office Hours in Gates B02
  - Tuesday 9am – 10am
  - Thursday 9am – 10:30am

# Logistics

Course Overview

First Program

Excited?

Logistics

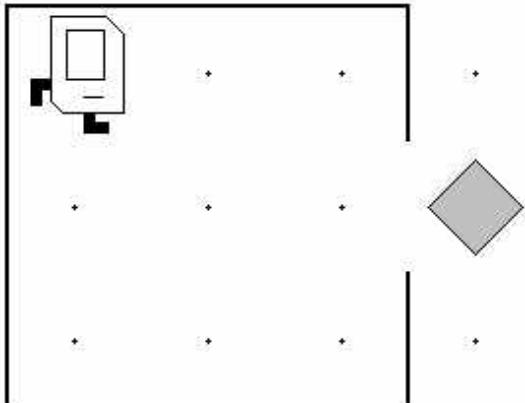
## Course Overview

First Program

Excited?

# What Was CS106A?

- CS 106A: Programming Methodology
  - a first course in **programming**, software development

A screenshot of the Eclipse IDE showing code coverage analysis for a Java project named "TestAPackages". The coverage report indicates 95.4% coverage across 149 lines of code. The code editor shows a snippet of Java code for a class named "CursorableLinkedList". The coverage details table is as follows:

What's next in Computer Science?



*[suspense]*

CS106B!

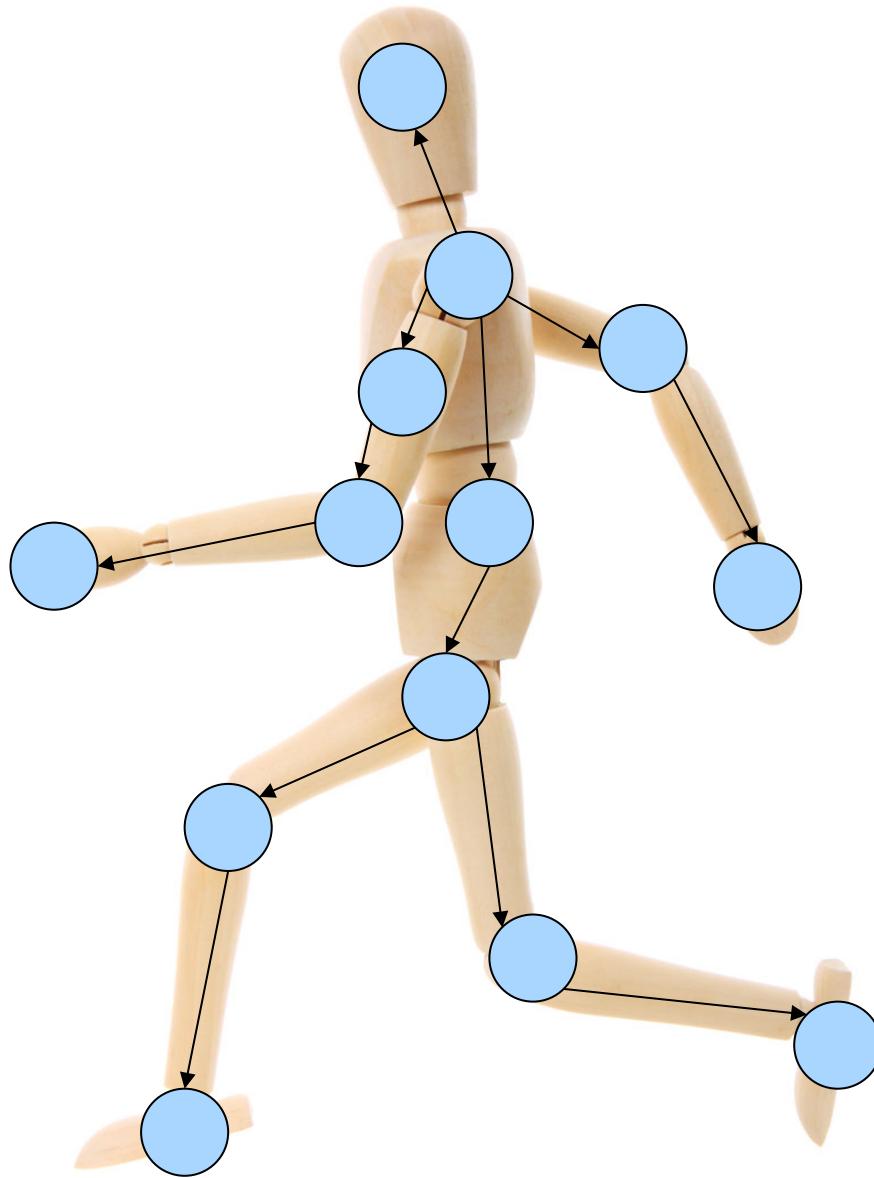
# Goals for the Course

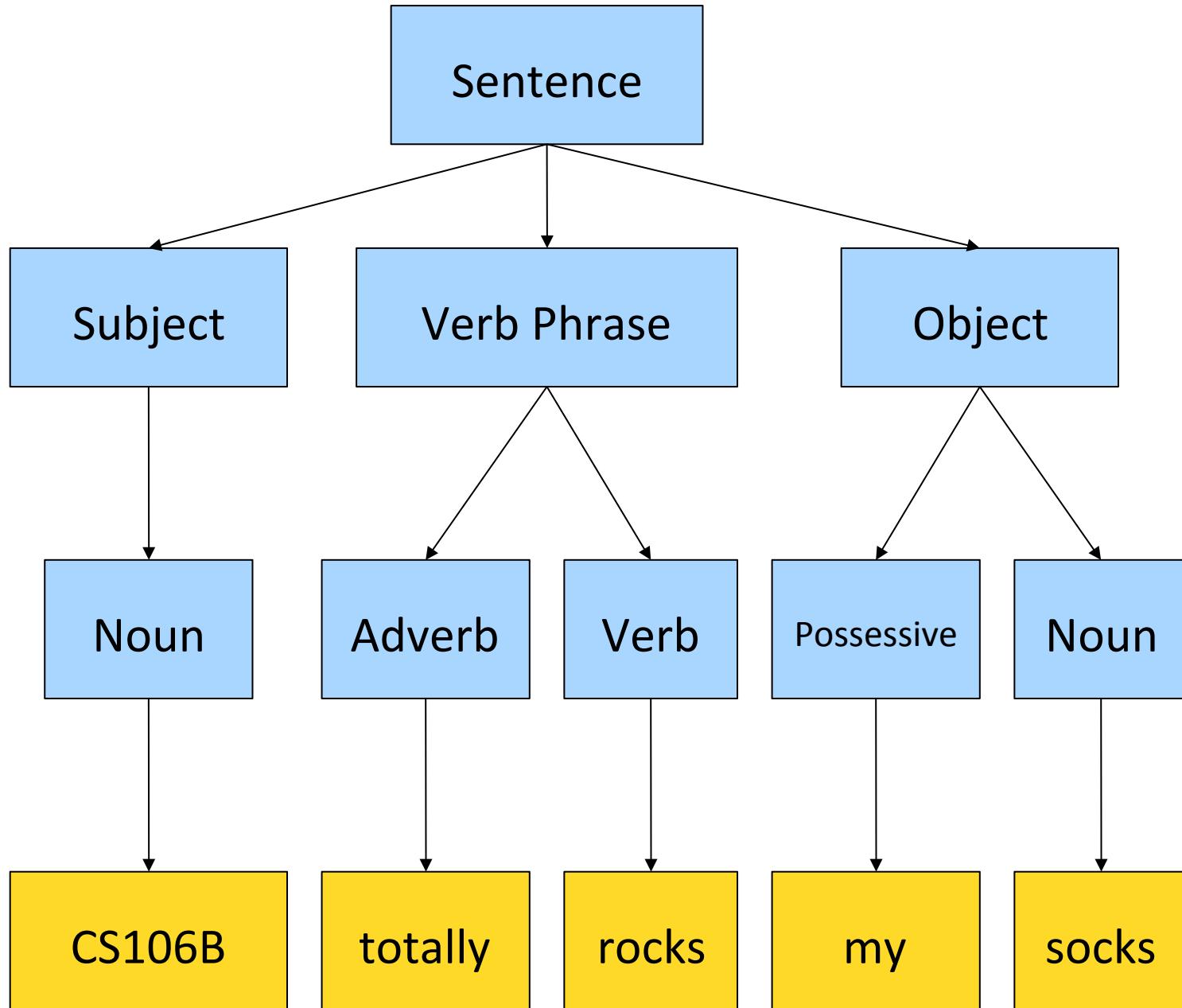
- **Learn how to model and solve complex problems with computers.**
- To that end:
  - Explore common abstractions for representing problems.
  - Harness recursion and understand how to think about problems recursively.
  - Learn and analyze different approaches for solving problems.

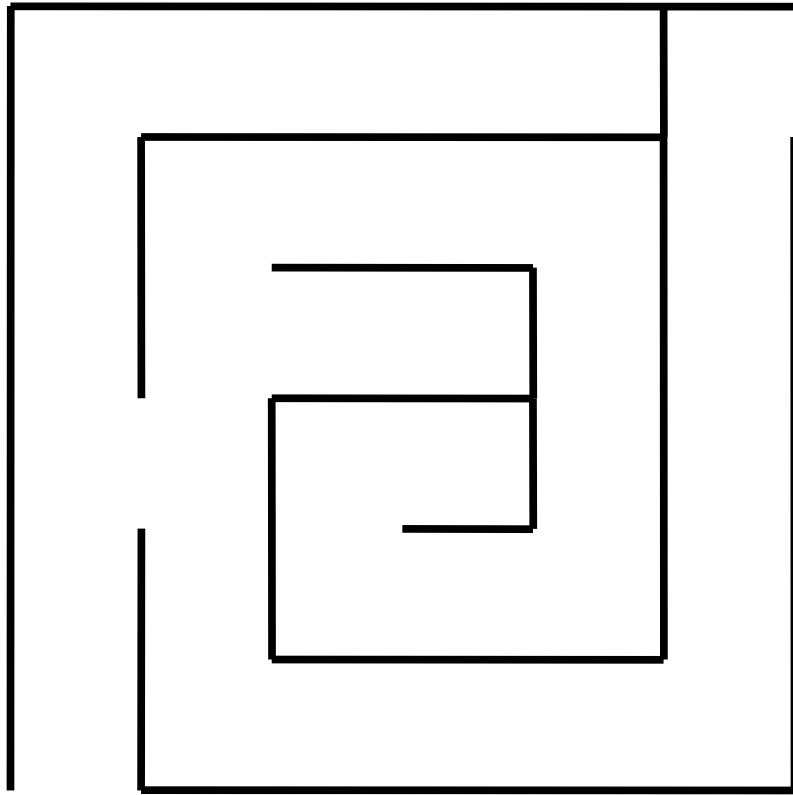
# Goals for the Course

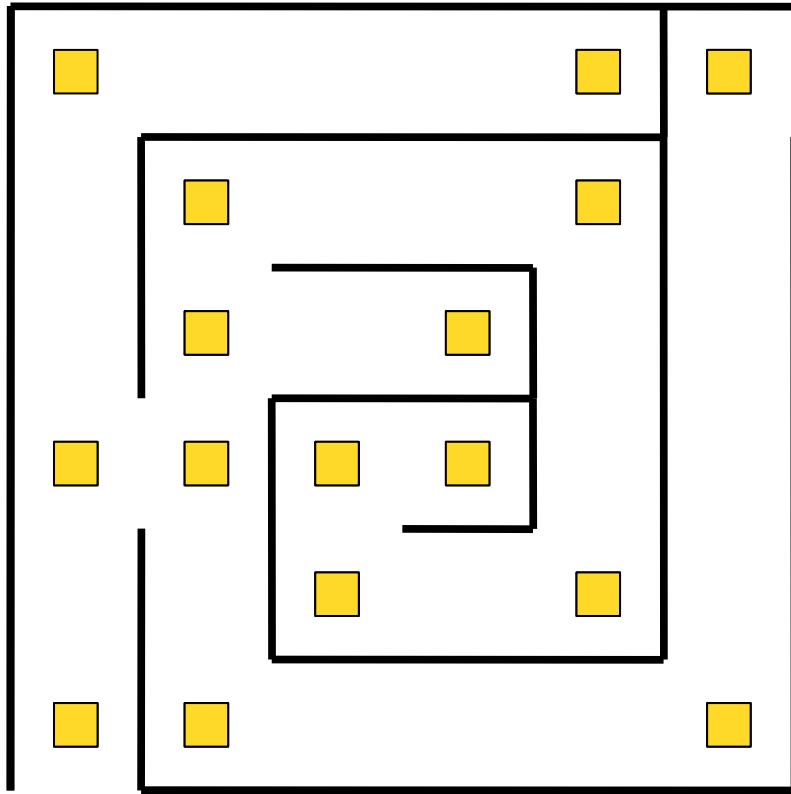
- Learn how to model and solve complex problems with computers.
- To that end:
  - **Explore common abstractions for representing problems.**
  - Harness recursion and understand how to think about problems recursively.
  - Learn and analyze different approaches for solving problems.

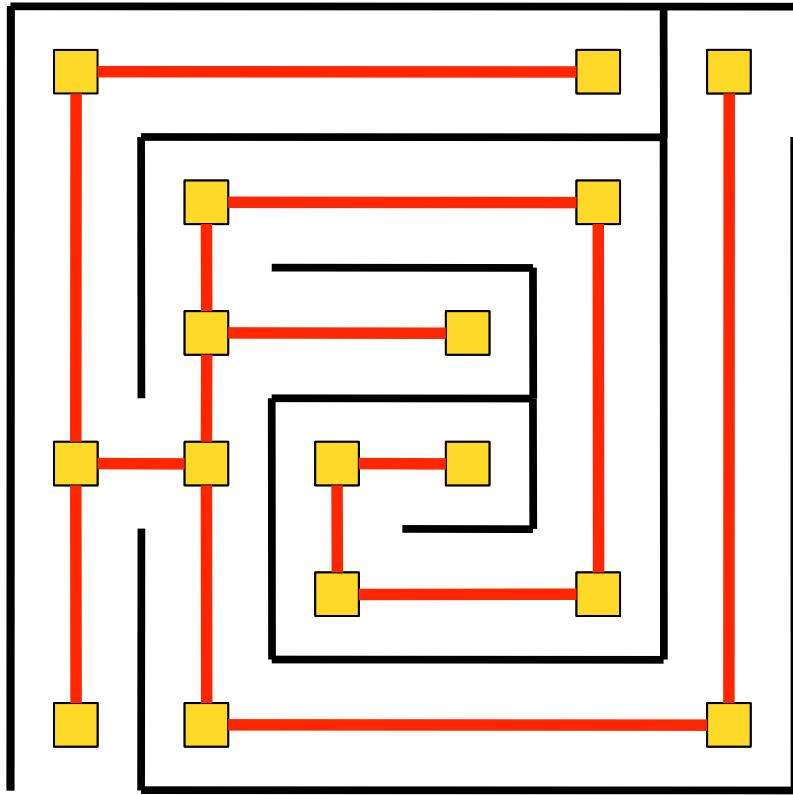


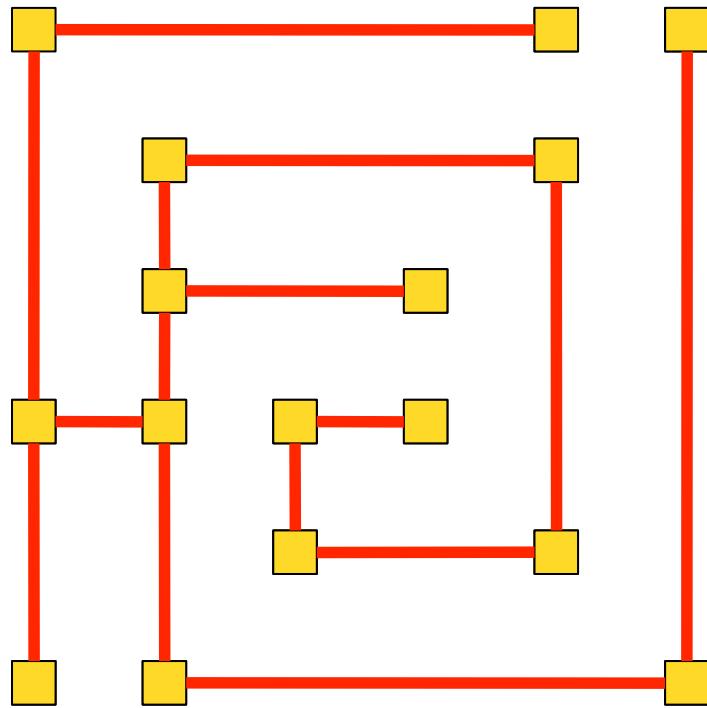




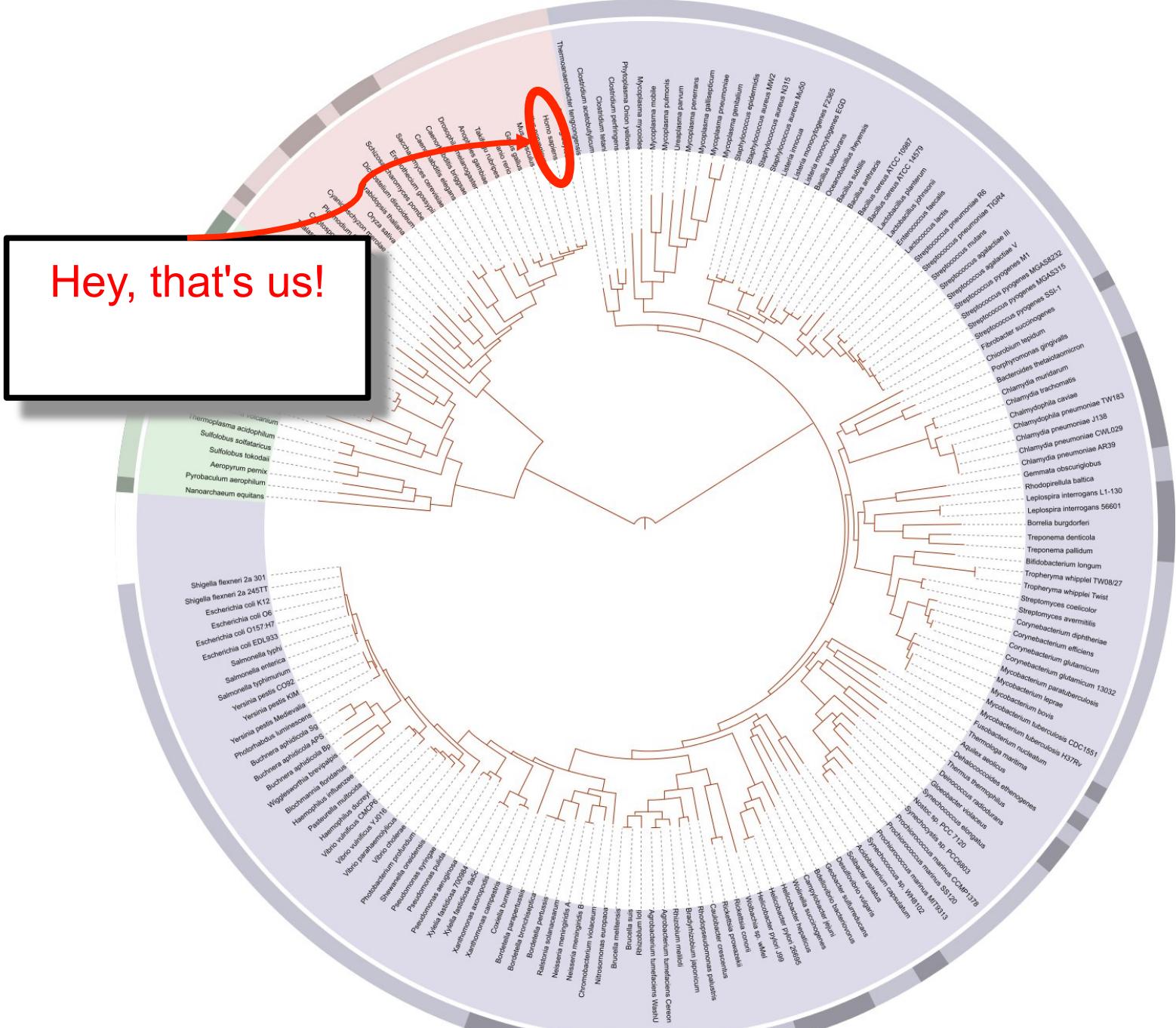


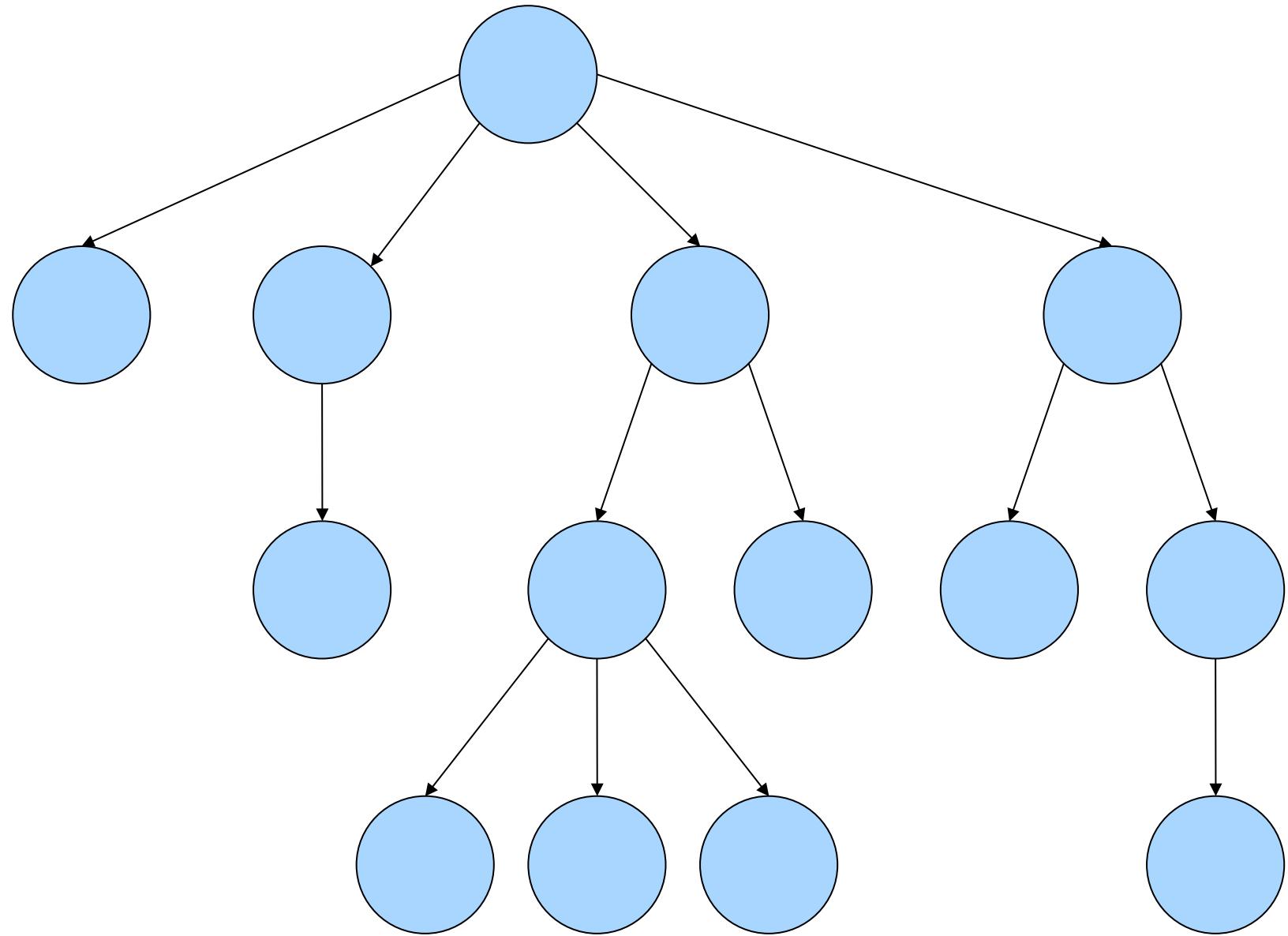






# Hey, that's us!





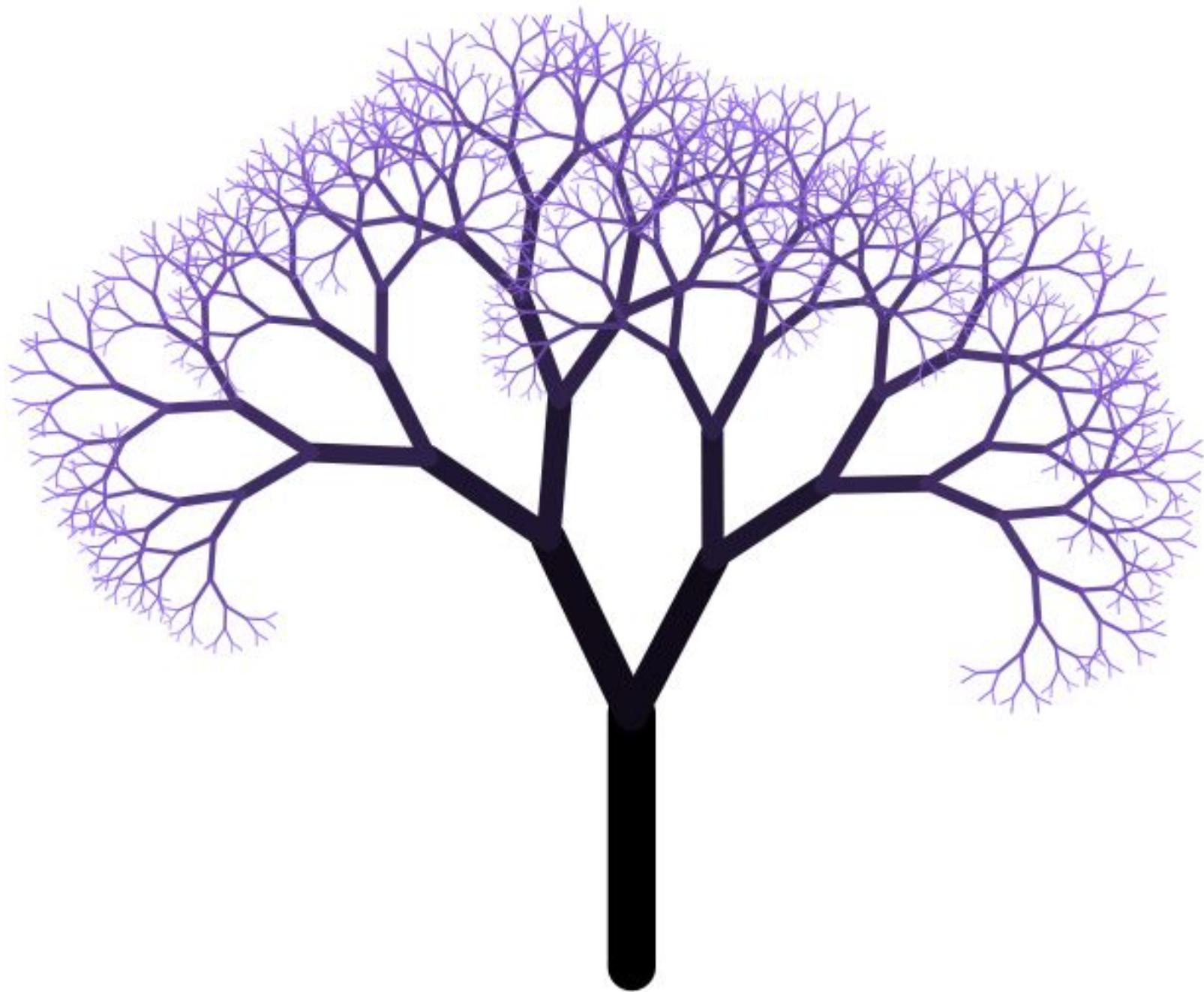
Building a vocabulary of **abstractions**  
makes it possible to represent and solve a  
wider class of problems.

# Goals for the Course

- Learn how to model and solve complex problems with computers.
- To that end:
  - **Explore common abstractions for representing problems.**
  - Harness recursion and understand how to think about problems recursively.
  - Learn and analyze different approaches for solving problems.

# Goals for the Course

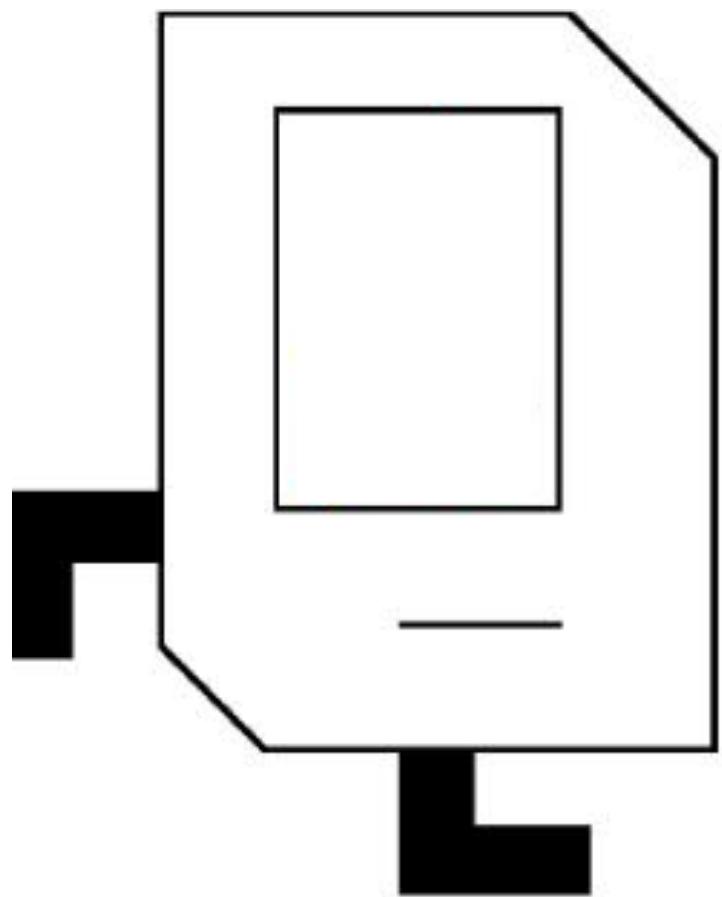
- Learn how to model and solve complex problems with computers.
- To that end:
  - Explore common abstractions for representing problems.
  - **Harness recursion and understand how to think about problems recursively.**
  - Learn and analyze different approaches for solving problems.

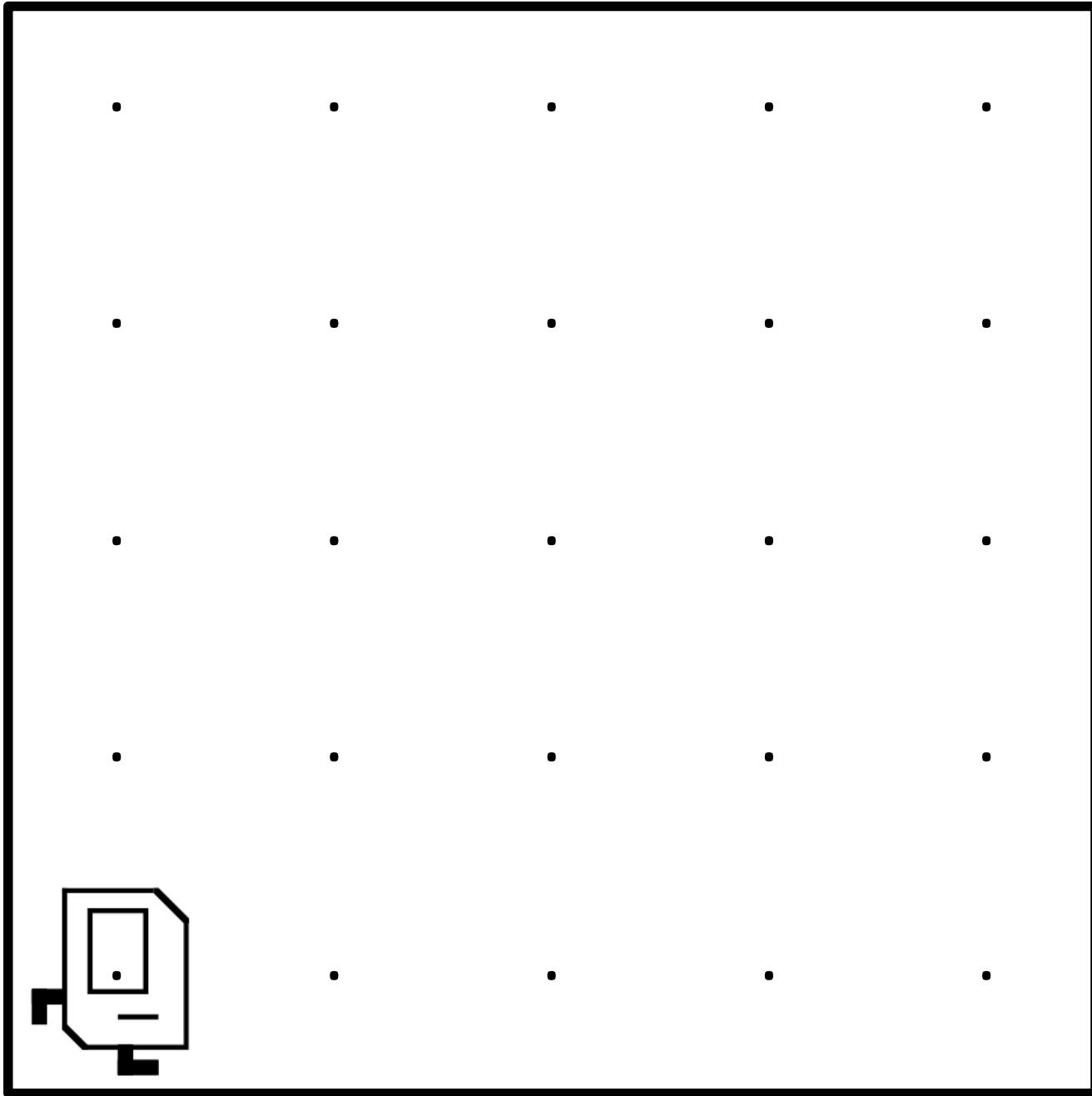


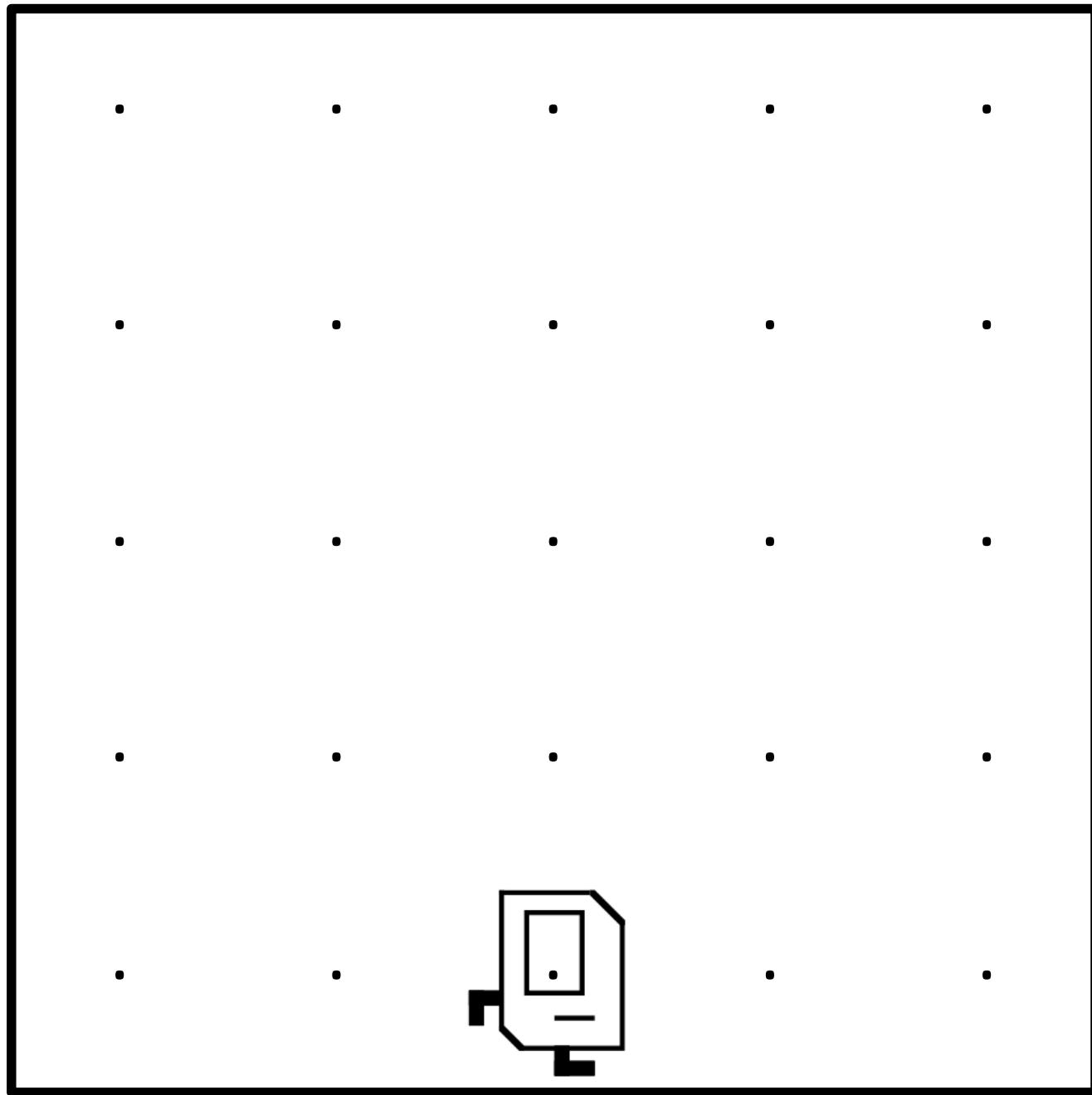


富嶽三十六景 神奈川沖  
浪裏

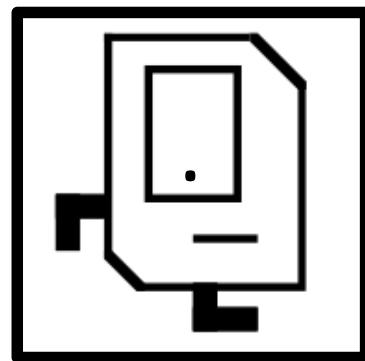
葛飾北斎

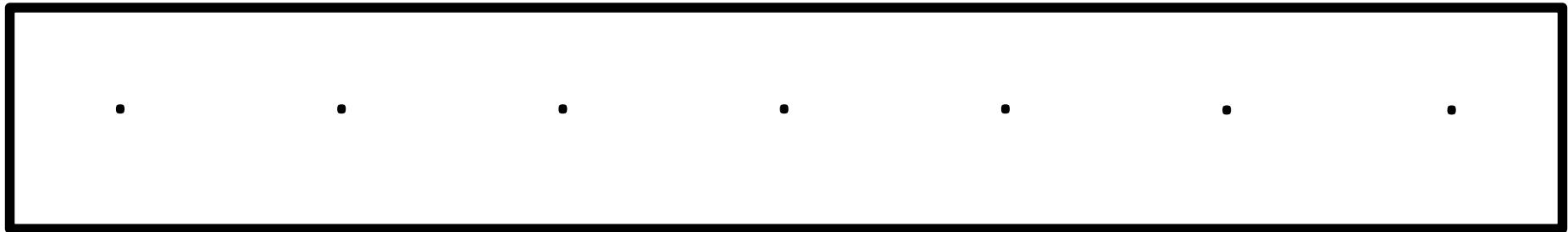


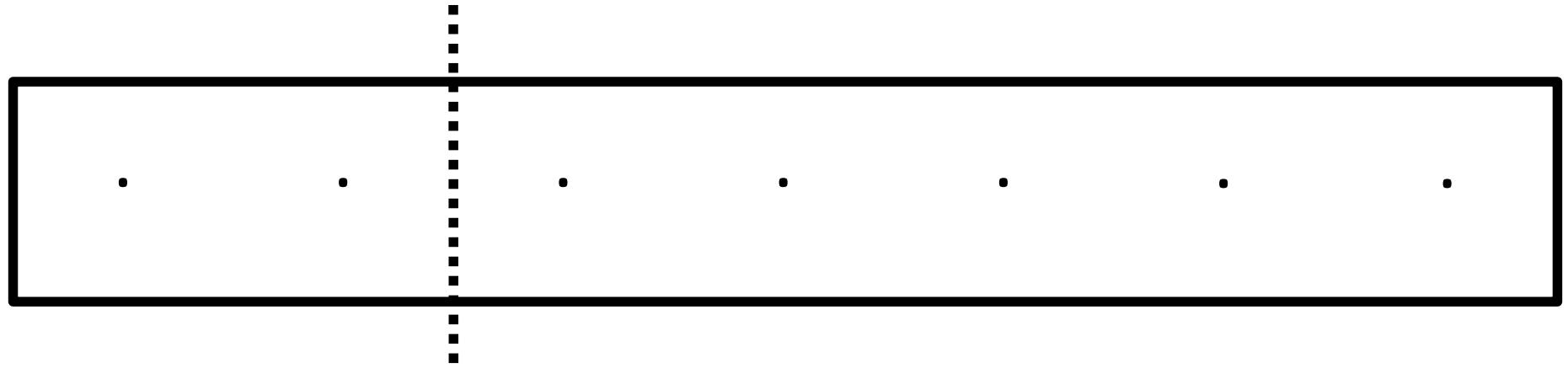


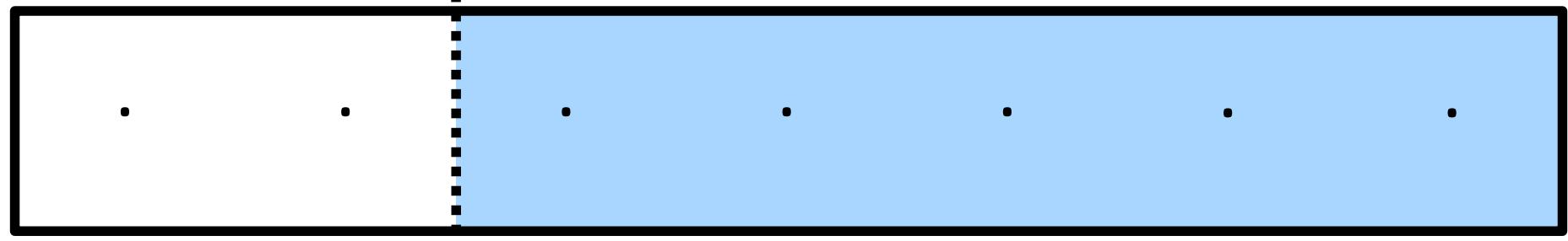


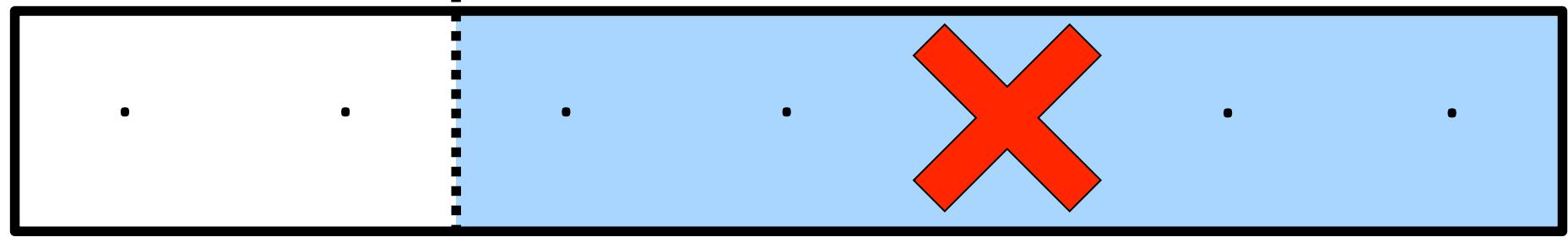
**Width 1**

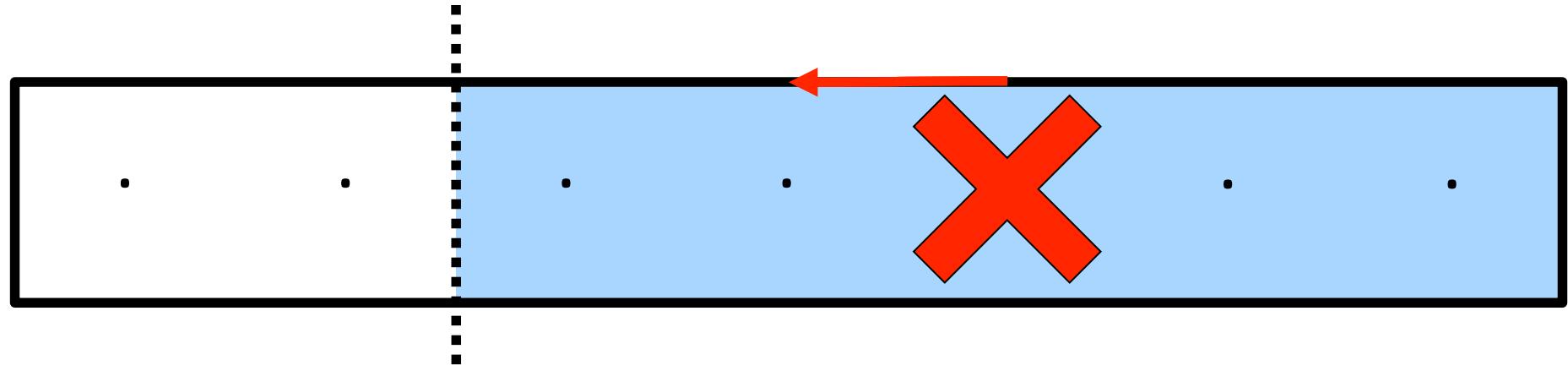


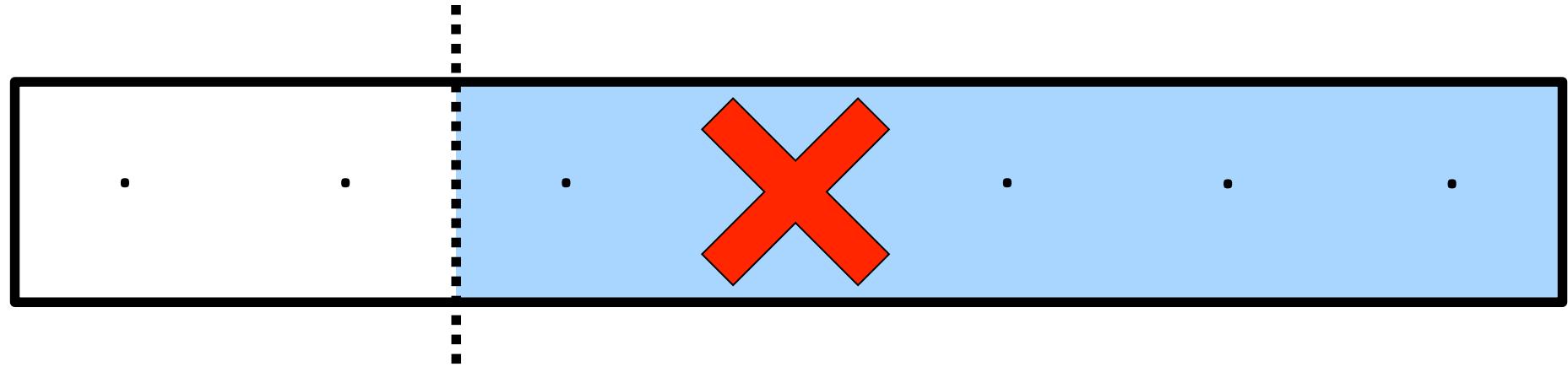


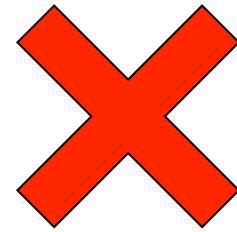












# Finding the Midpoint

- If the width is 1, Karel is standing on the midpoint.
- Otherwise:
  - Take two safe steps forward.
  - Find the midpoint of the rest of the world.
  - Take one step backward.

# A Surprisingly Short Solution

A **recursive solution** is a solution that is defined in terms of itself.

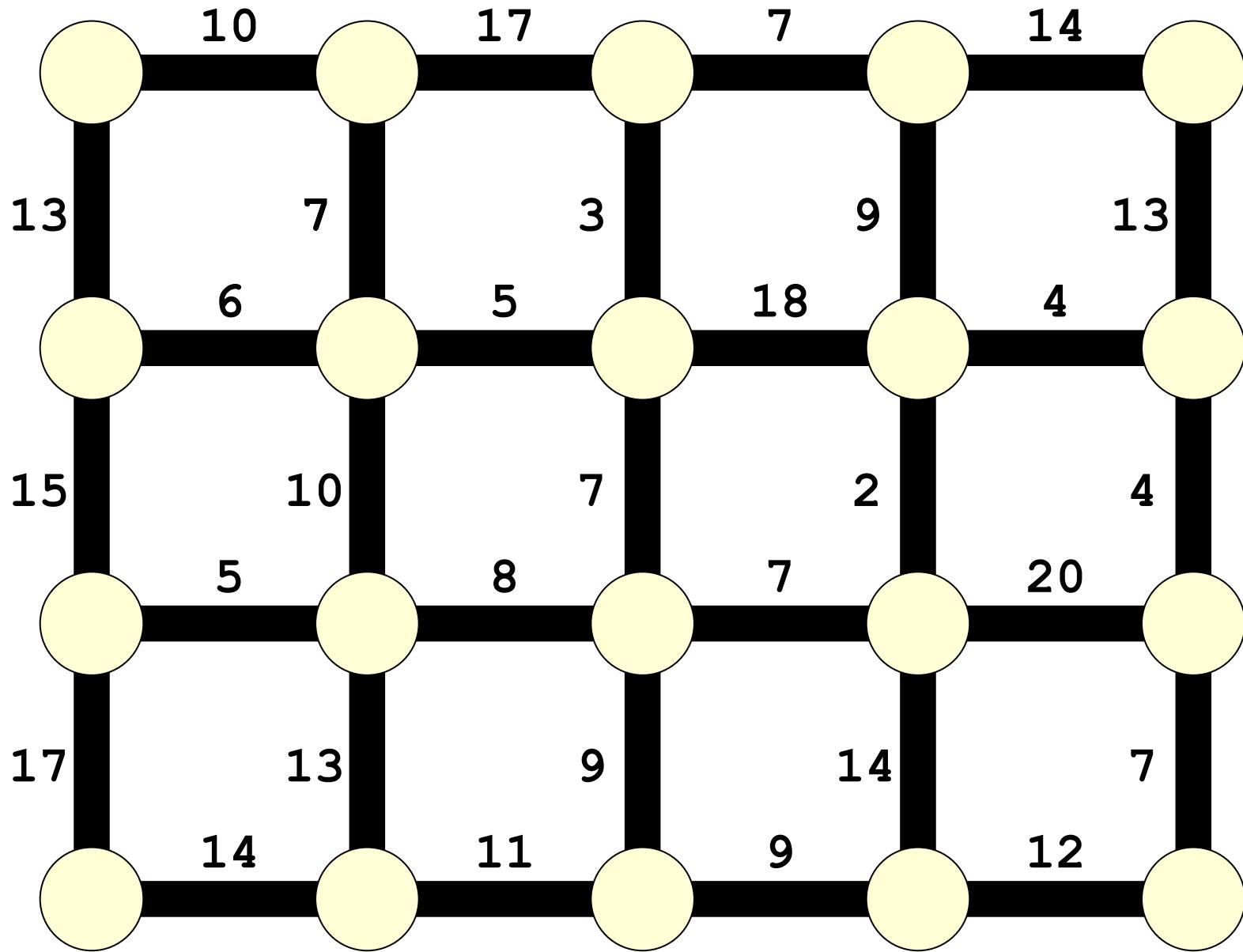
Thinking recursively allows you  
to solve an enormous class of  
problems cleanly and concisely.

# Goals for the Course

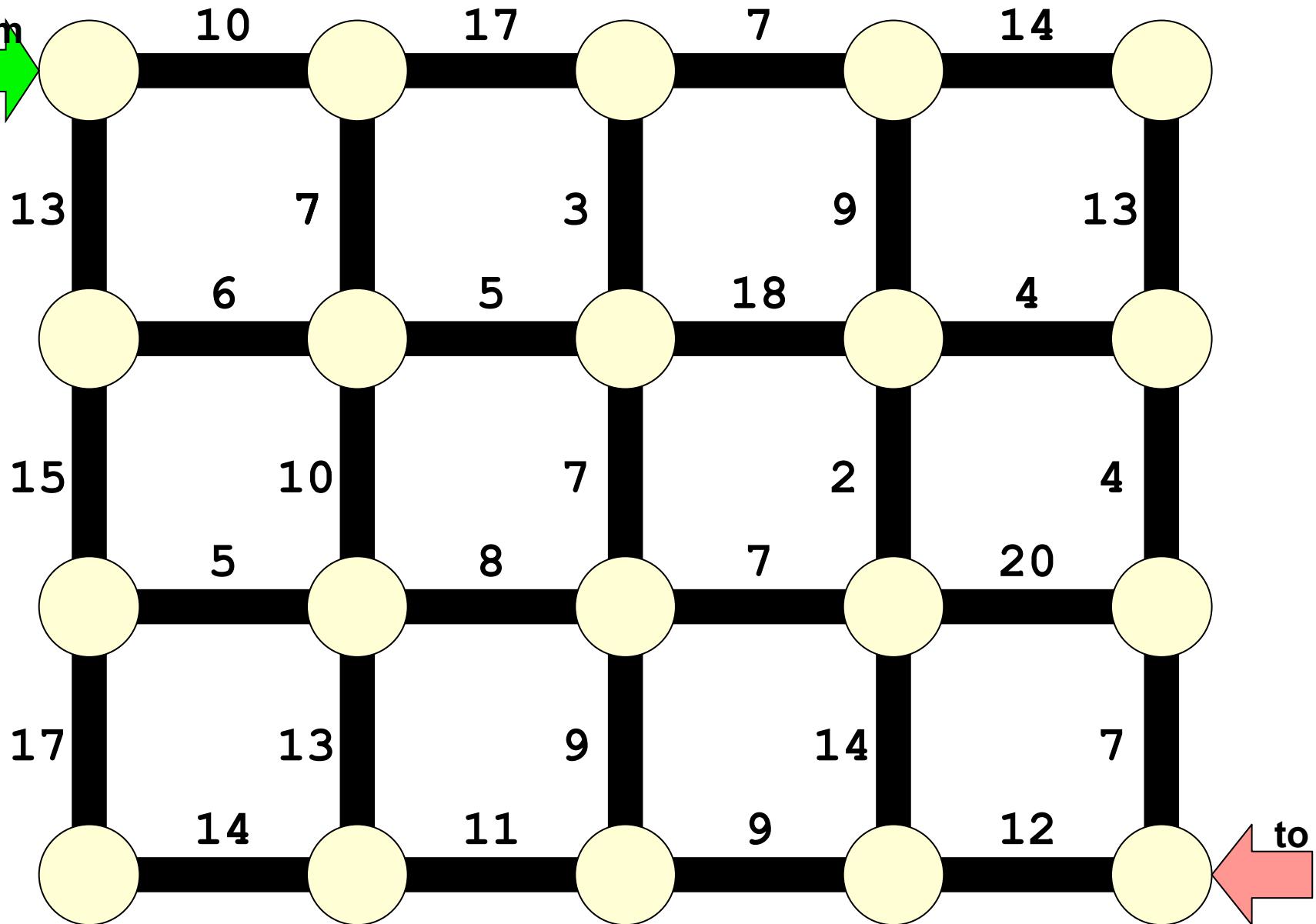
- Learn how to model and solve complex problems with computers.
- To that end:
  - Explore common abstractions for representing problems.
  - **Harness recursion and understand how to think about problems recursively.**
  - Learn and analyze different approaches for solving problems.

# Goals for the Course

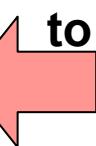
- Learn how to model and solve complex problems with computers.
- To that end:
  - Explore common abstractions for representing problems.
  - Harness recursion and understand how to think about problems recursively.
  - **Learn and analyze different approaches for solving problems.**

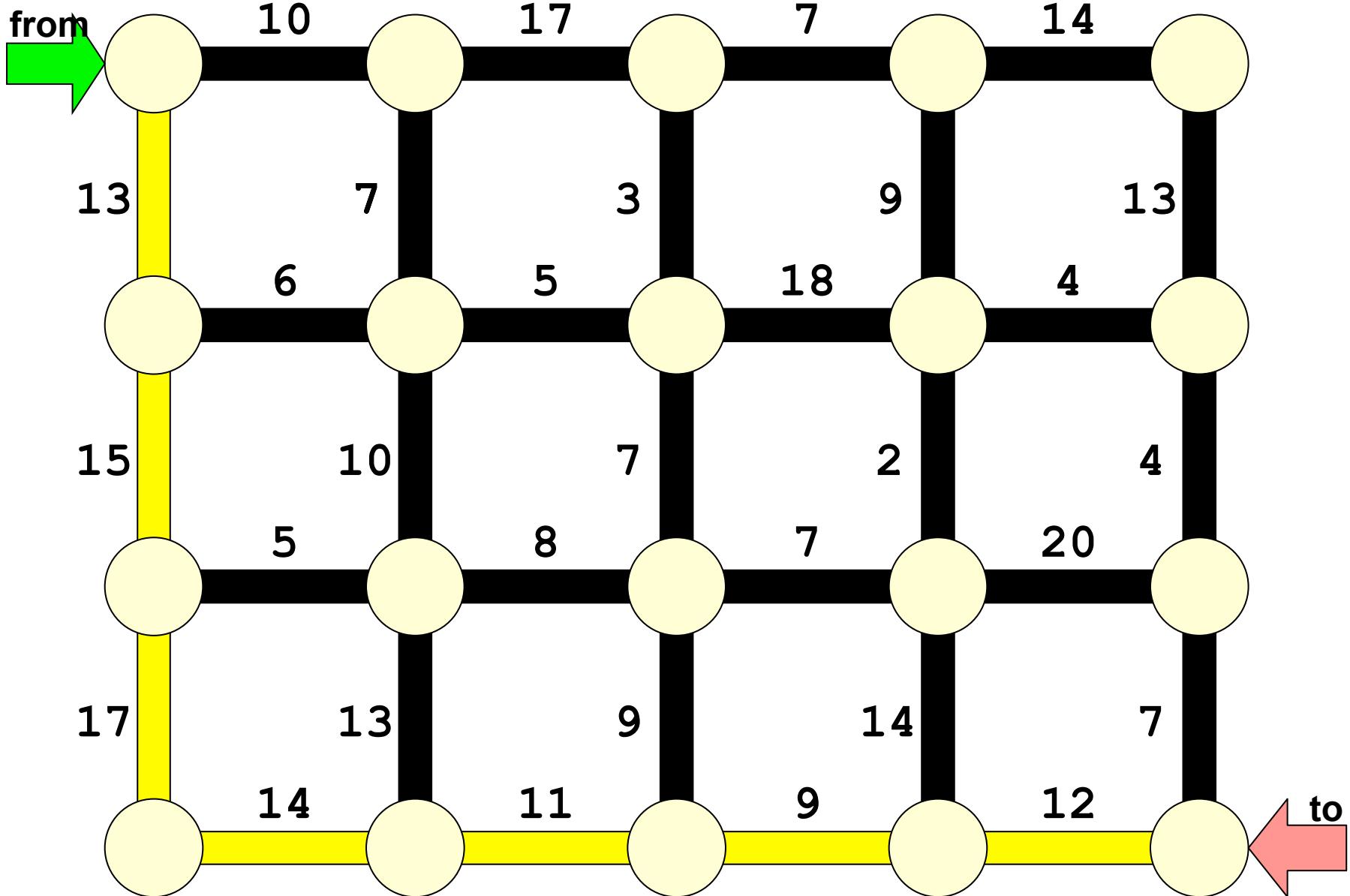


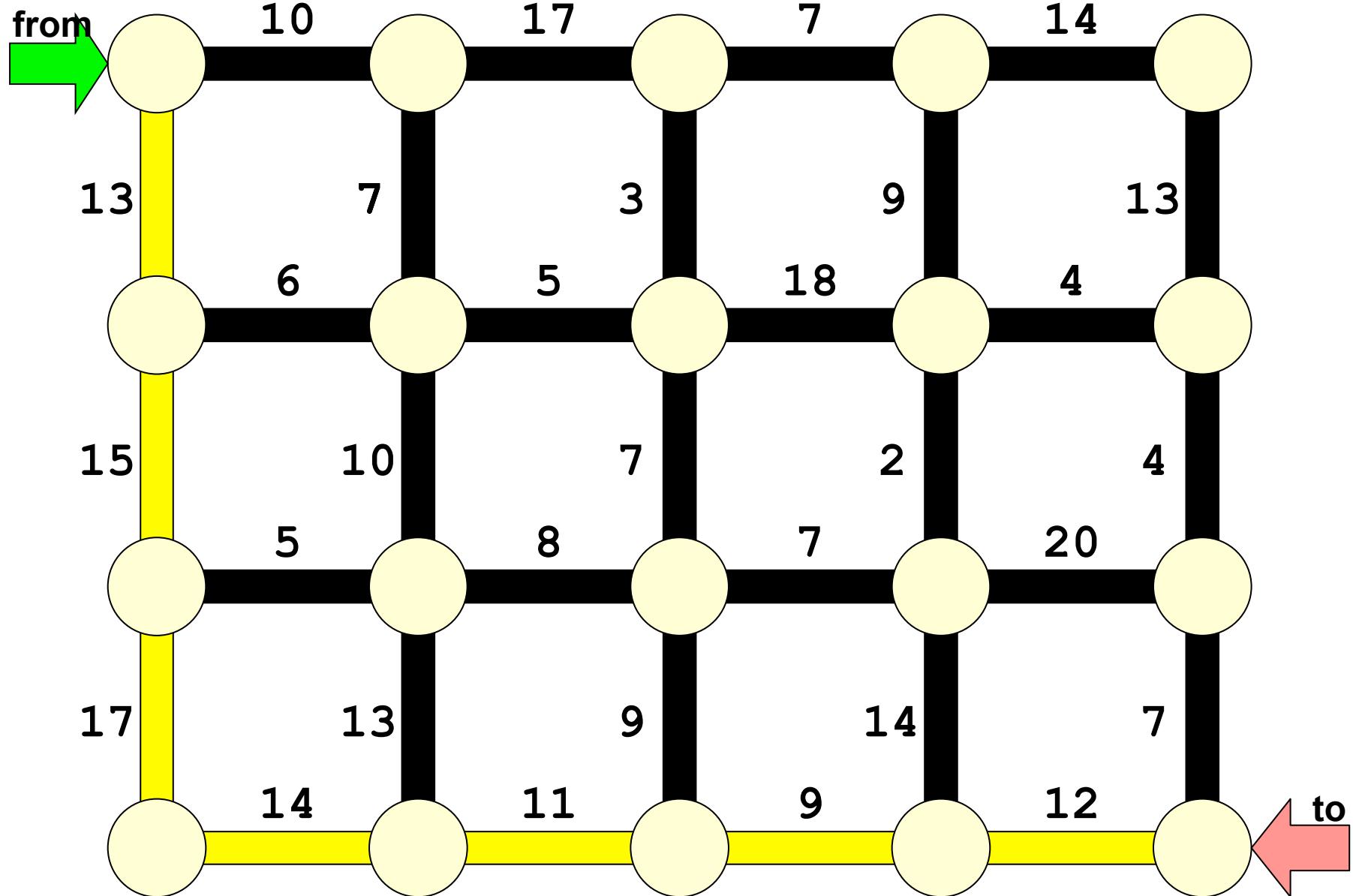
from



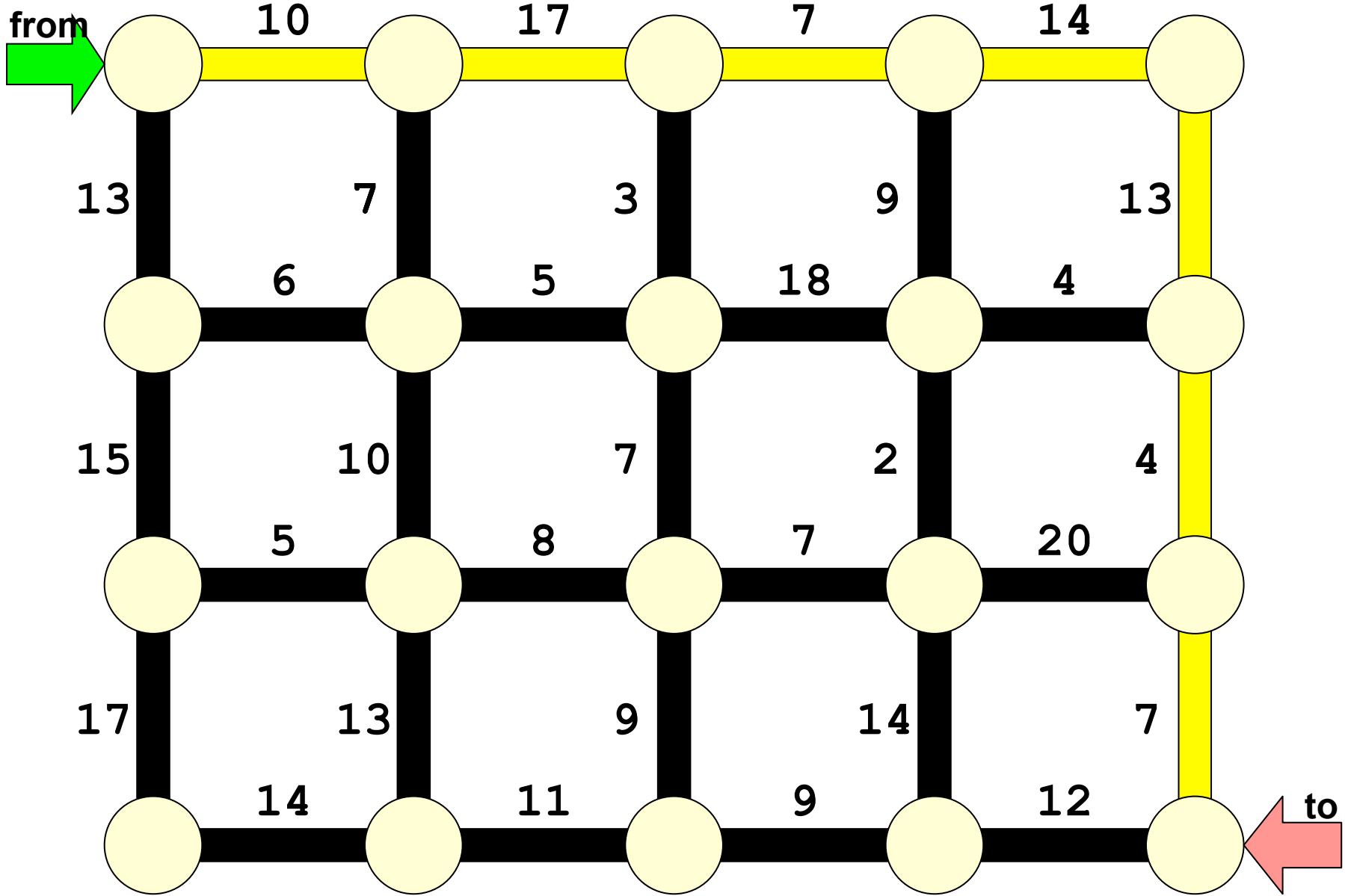
to

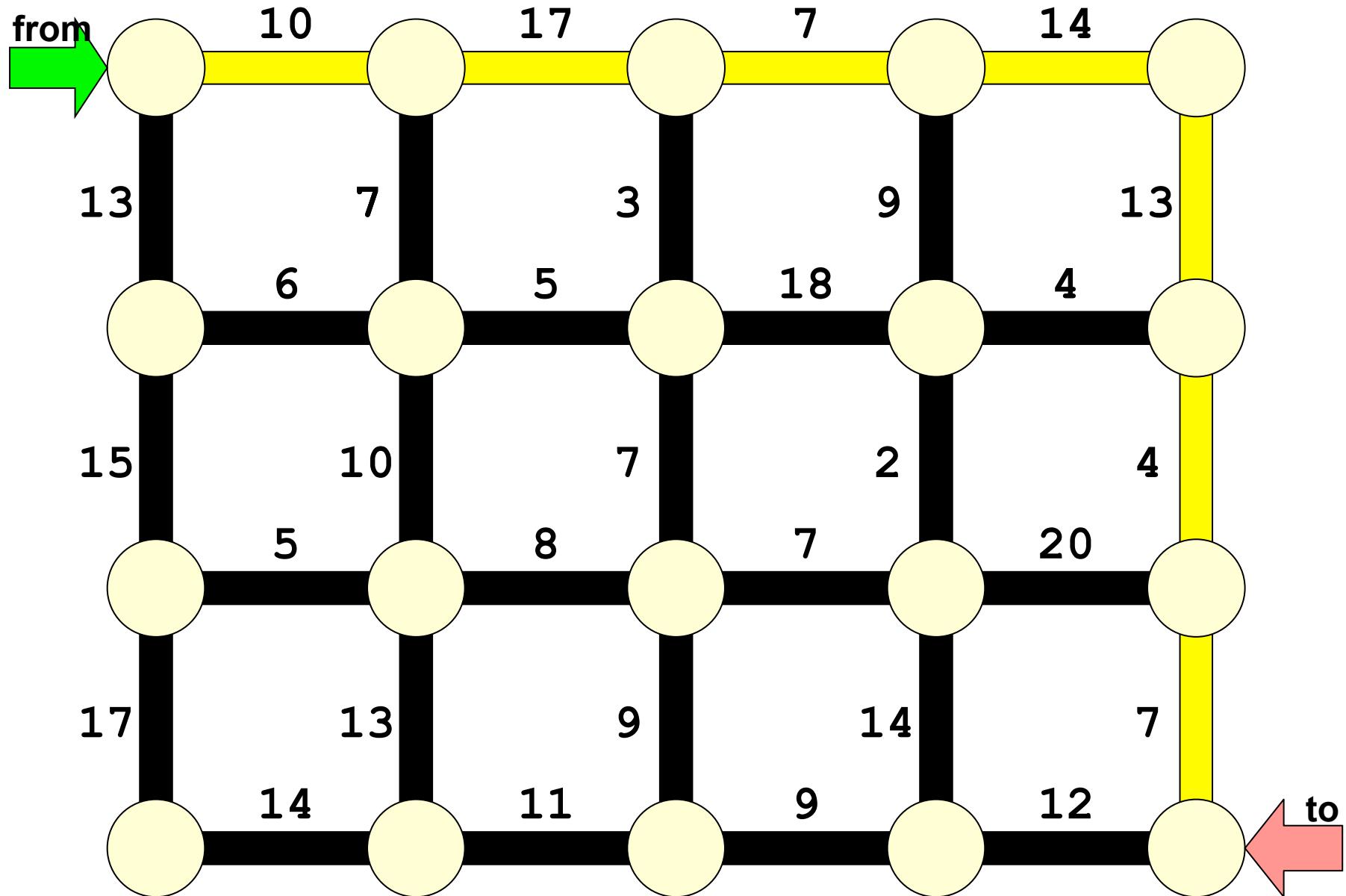




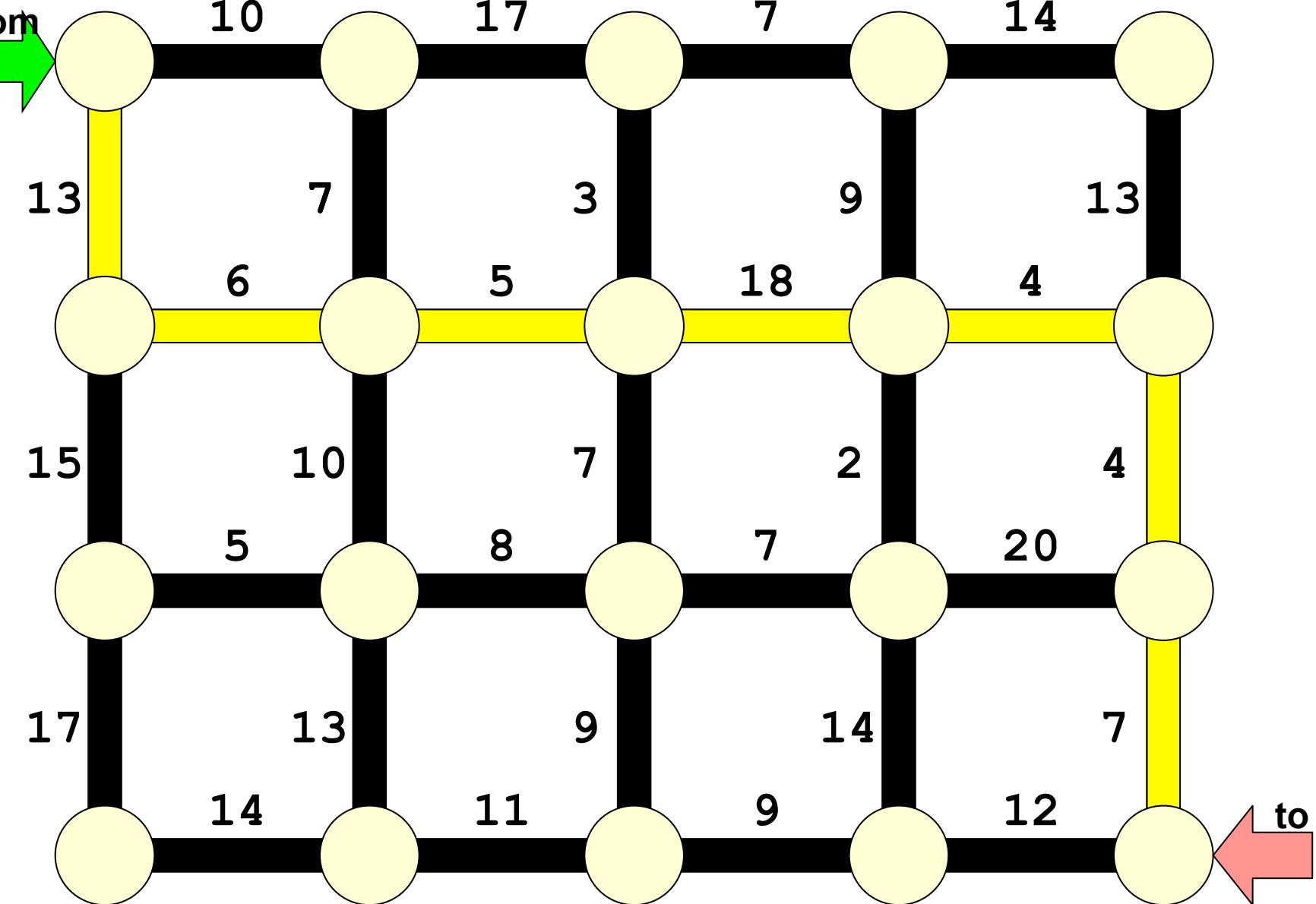


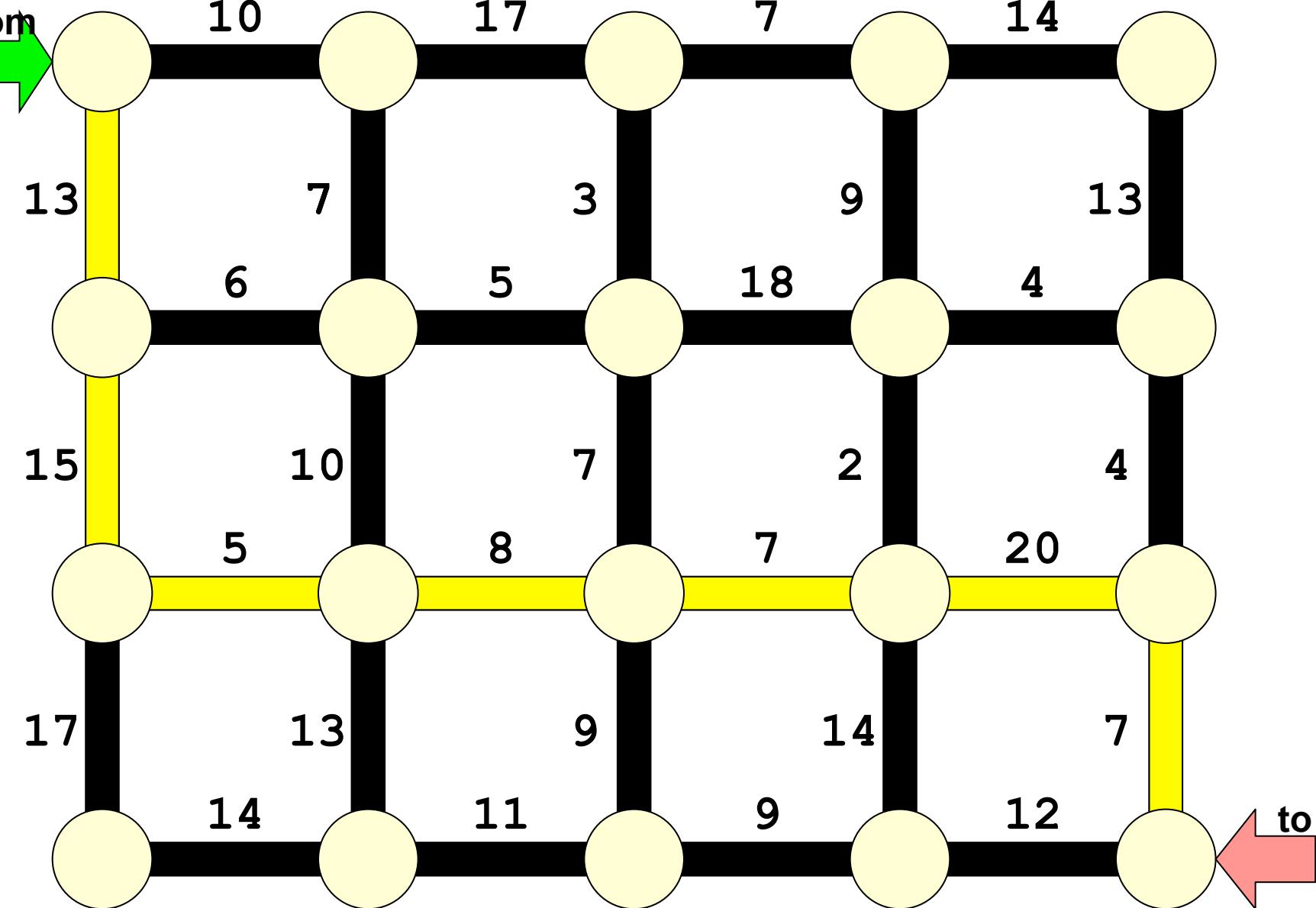
Travel Time:  $13 + 15 + 17 + 14 + 11 + 9 + 12 = \mathbf{91}$

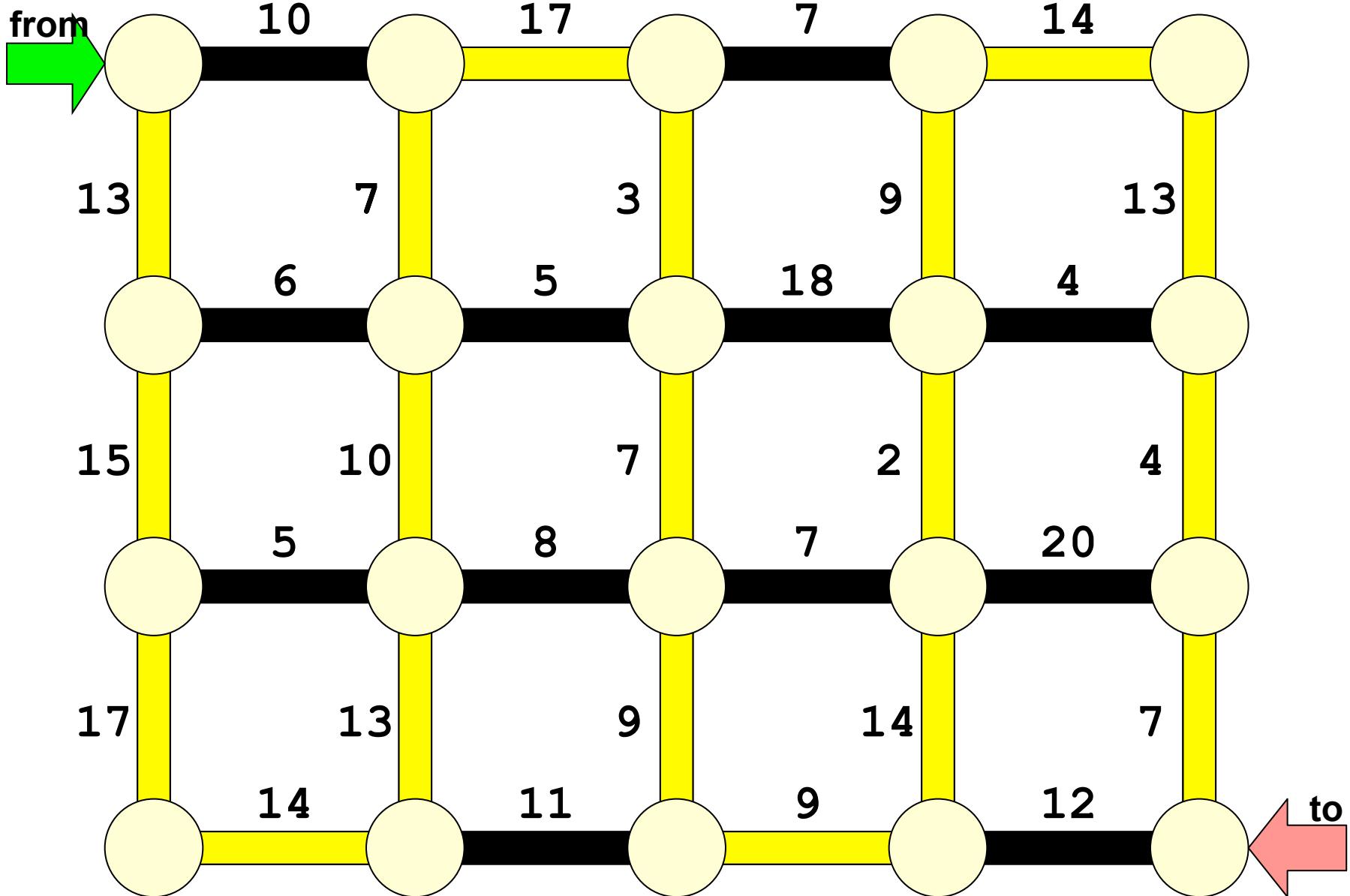


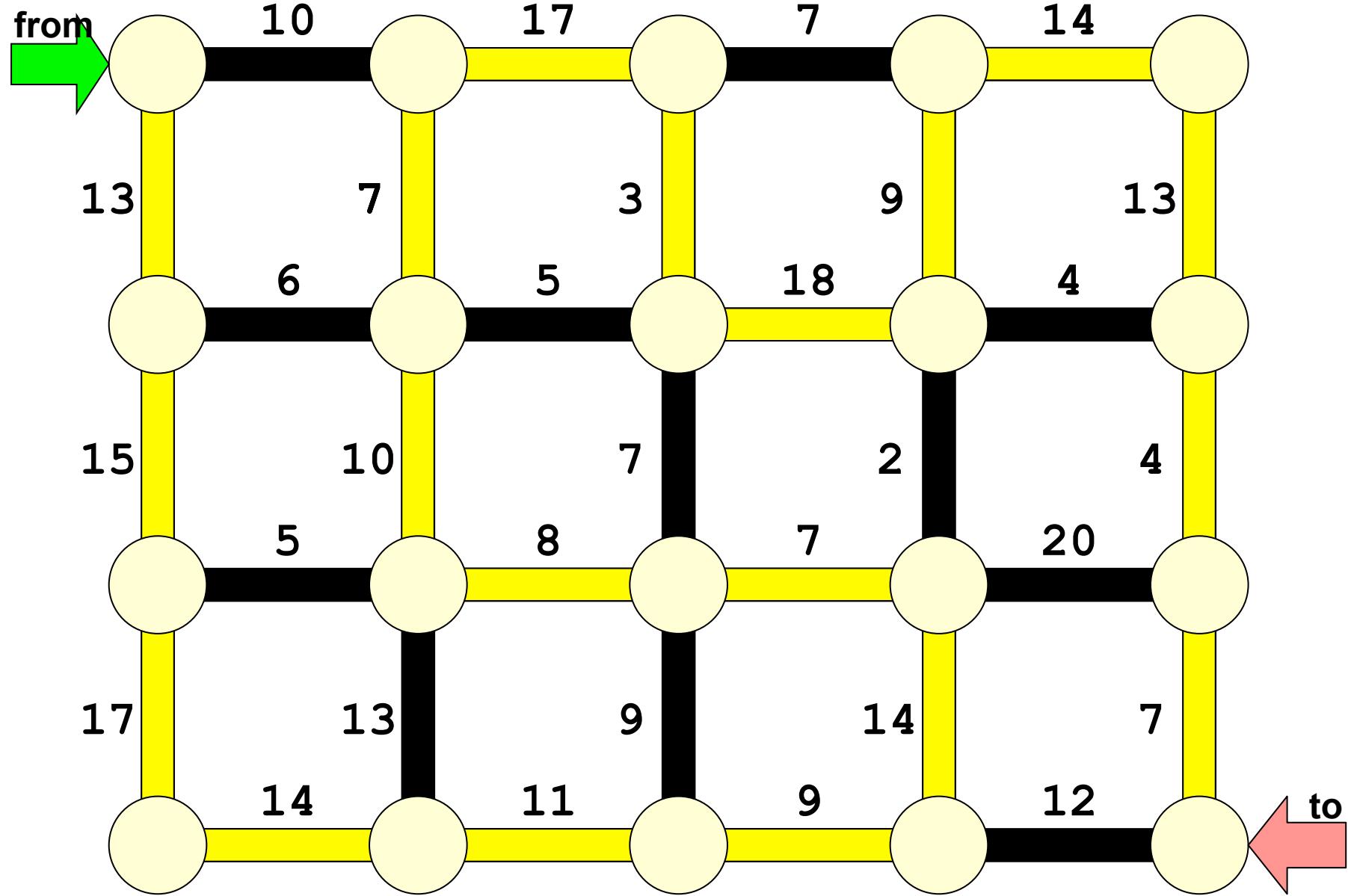


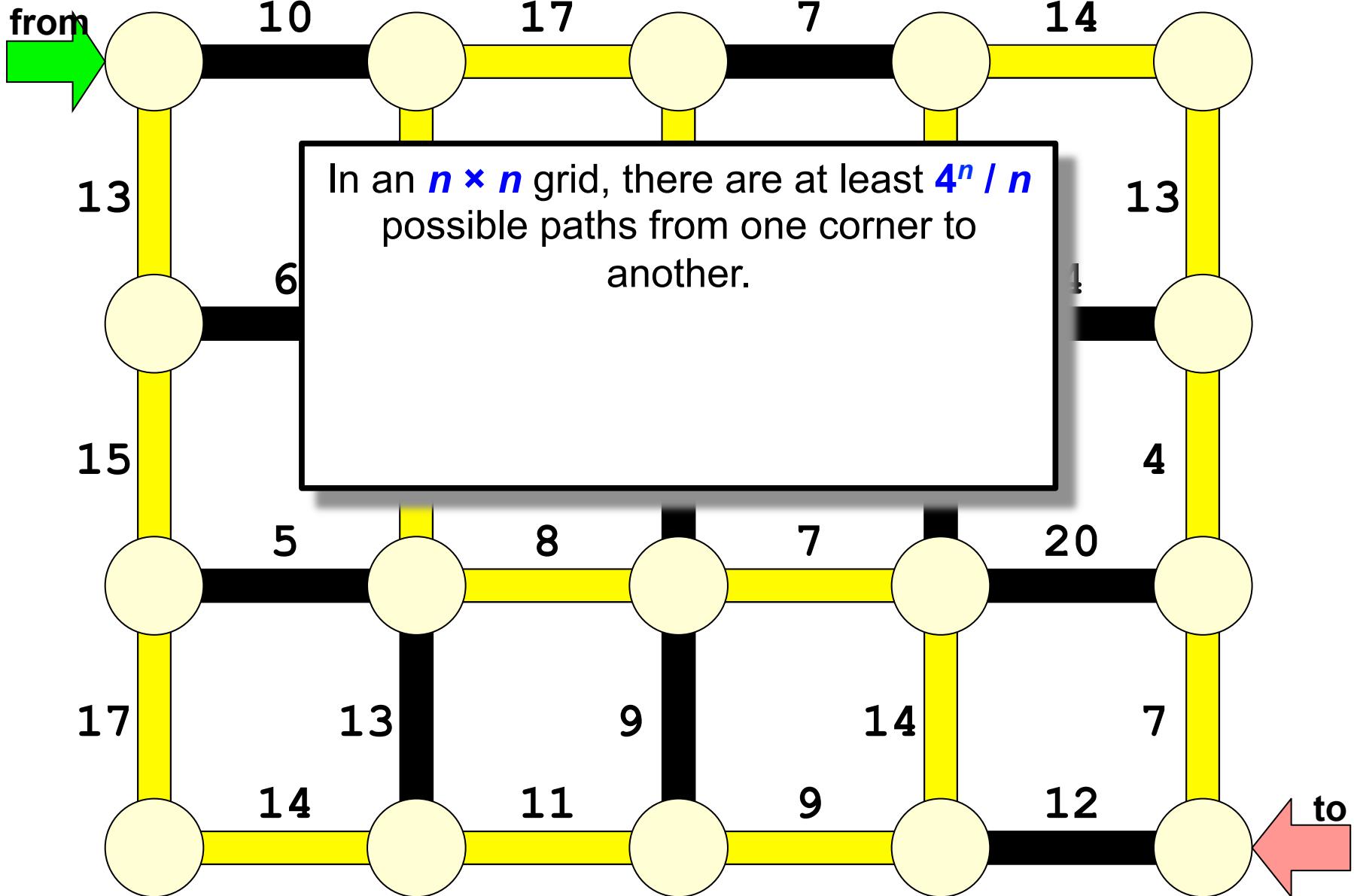
Travel Time:  $10 + 17 + 7 + 14 + 13 + 4 + 7 = \mathbf{72}$

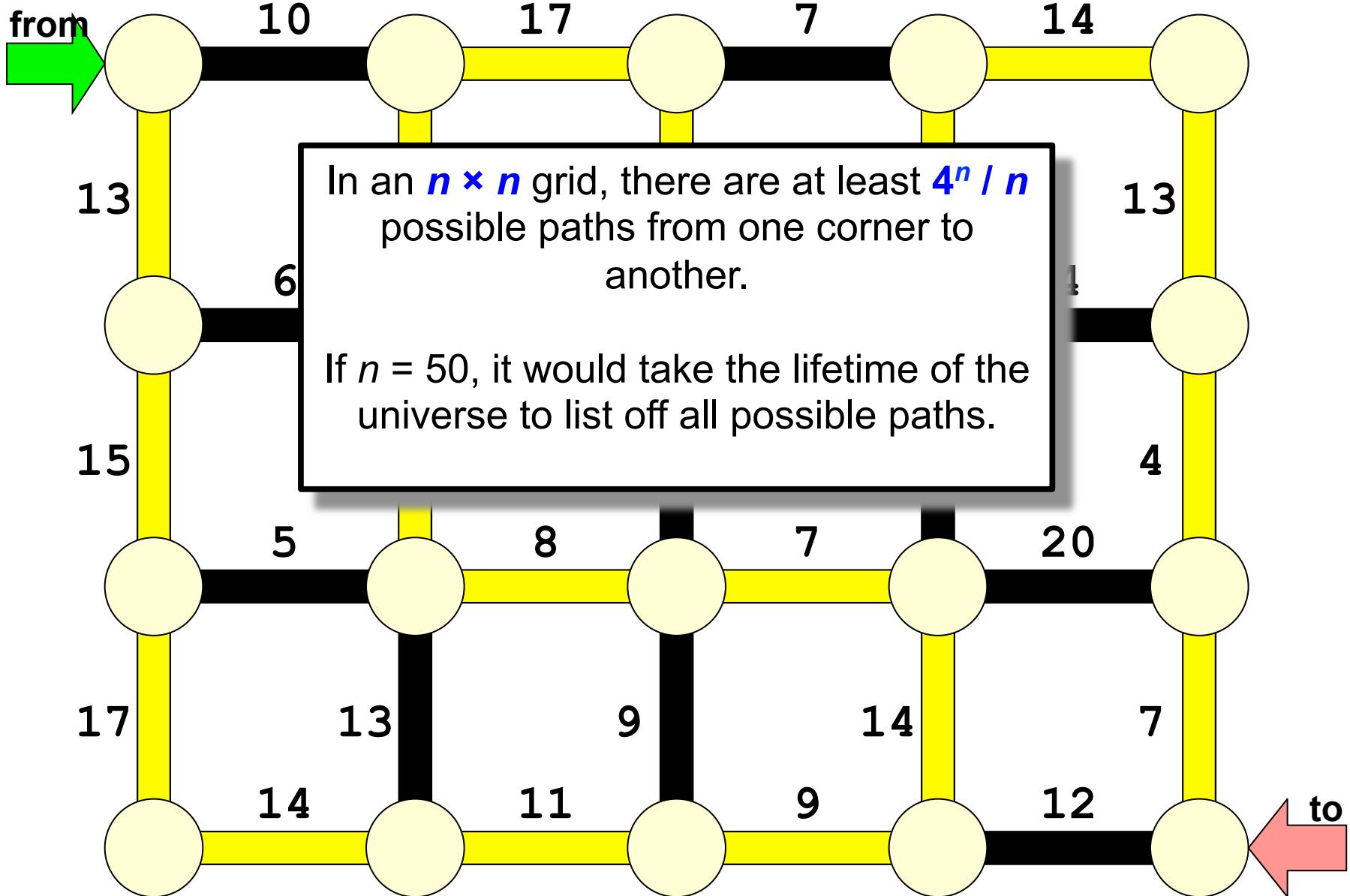


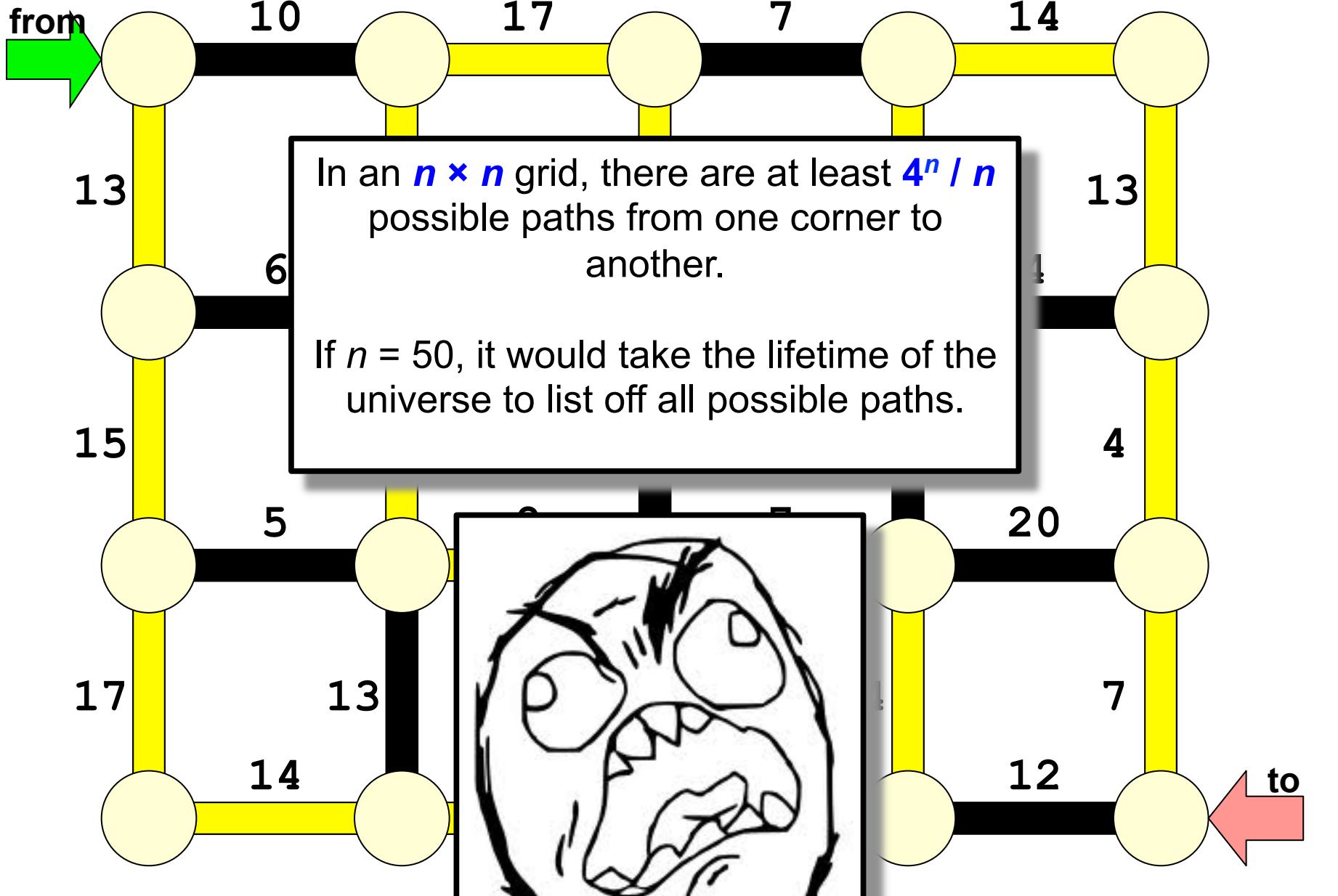




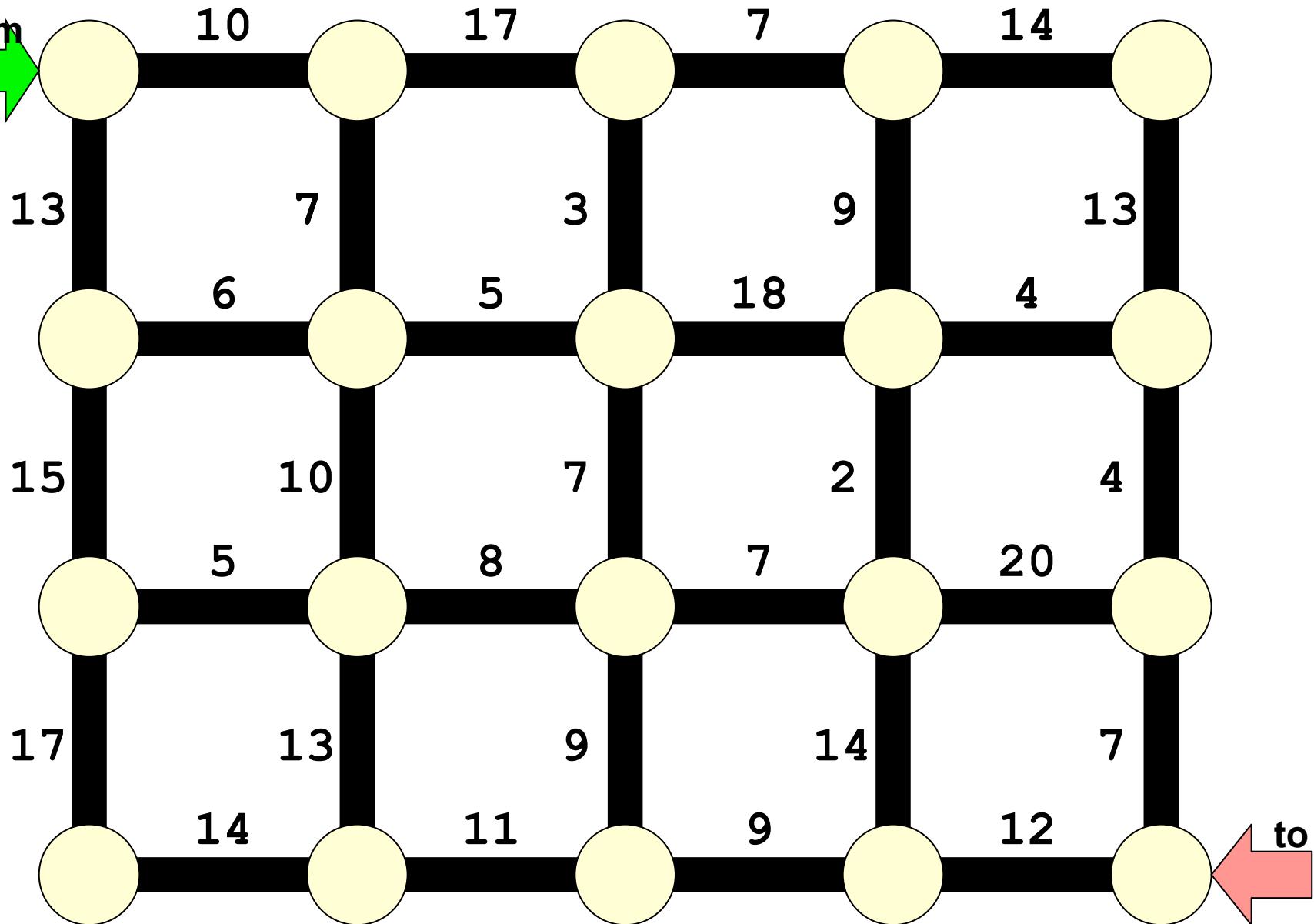




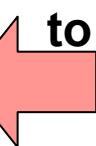




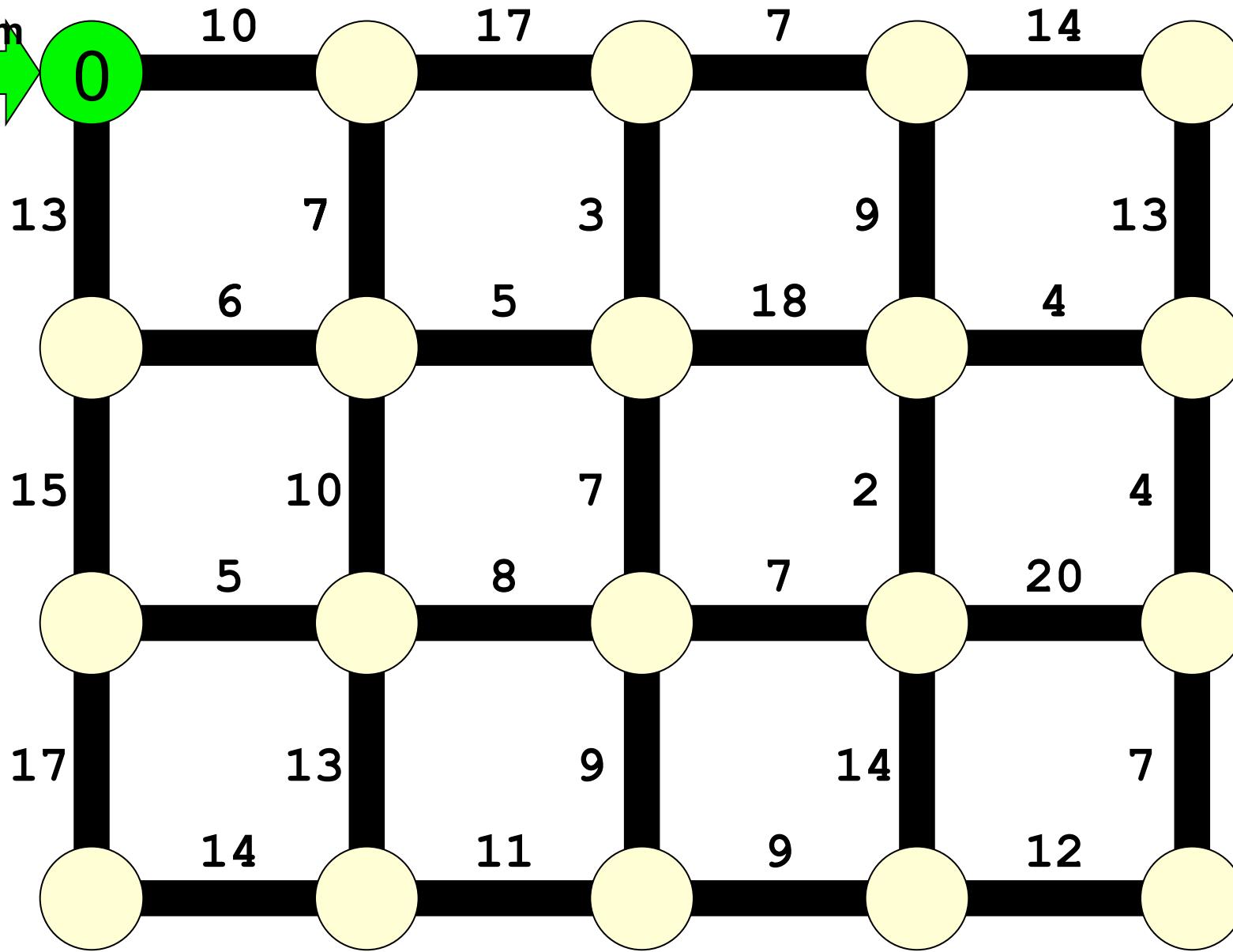
from



to

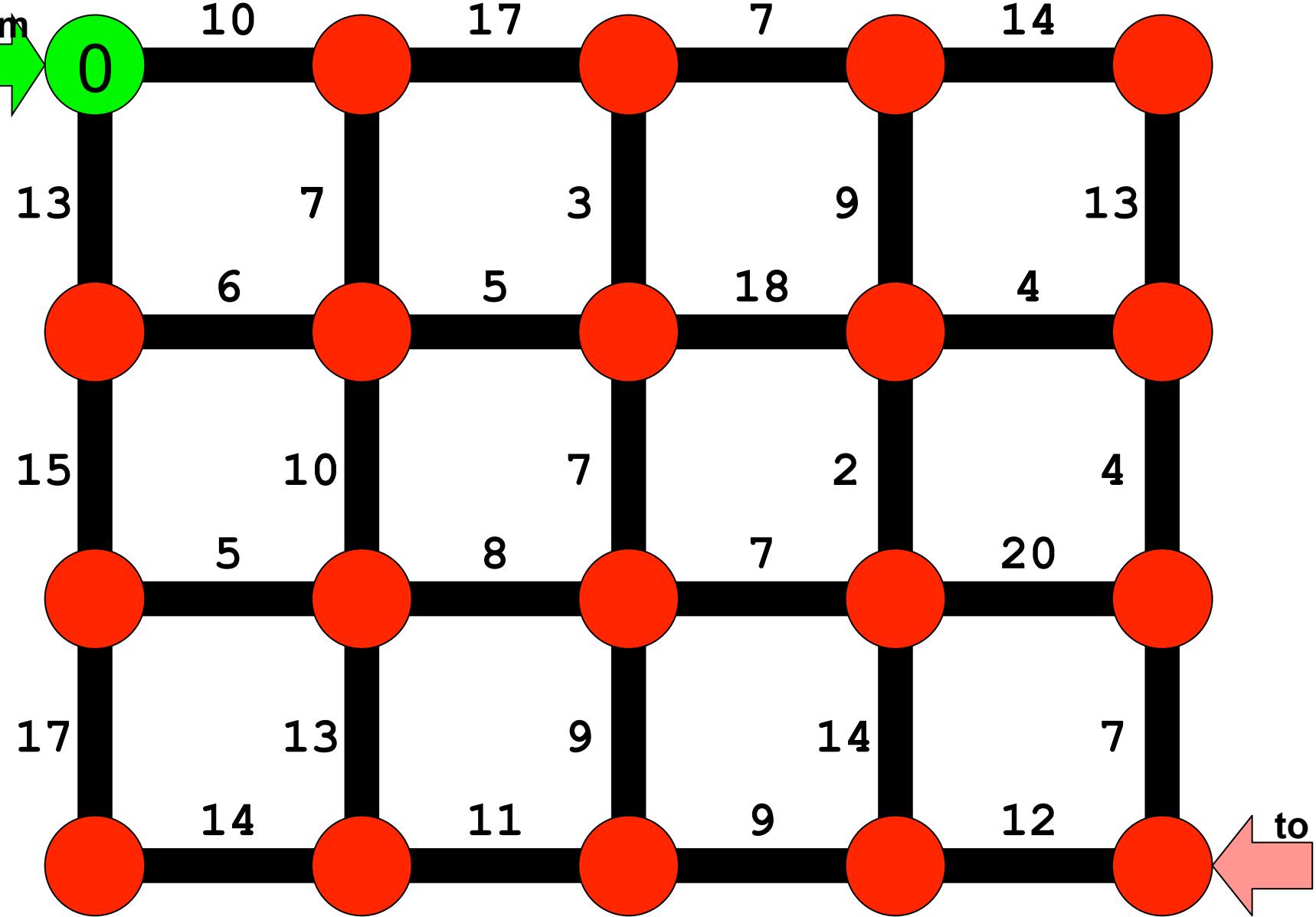
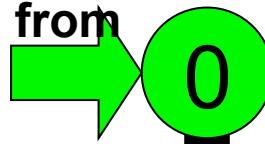


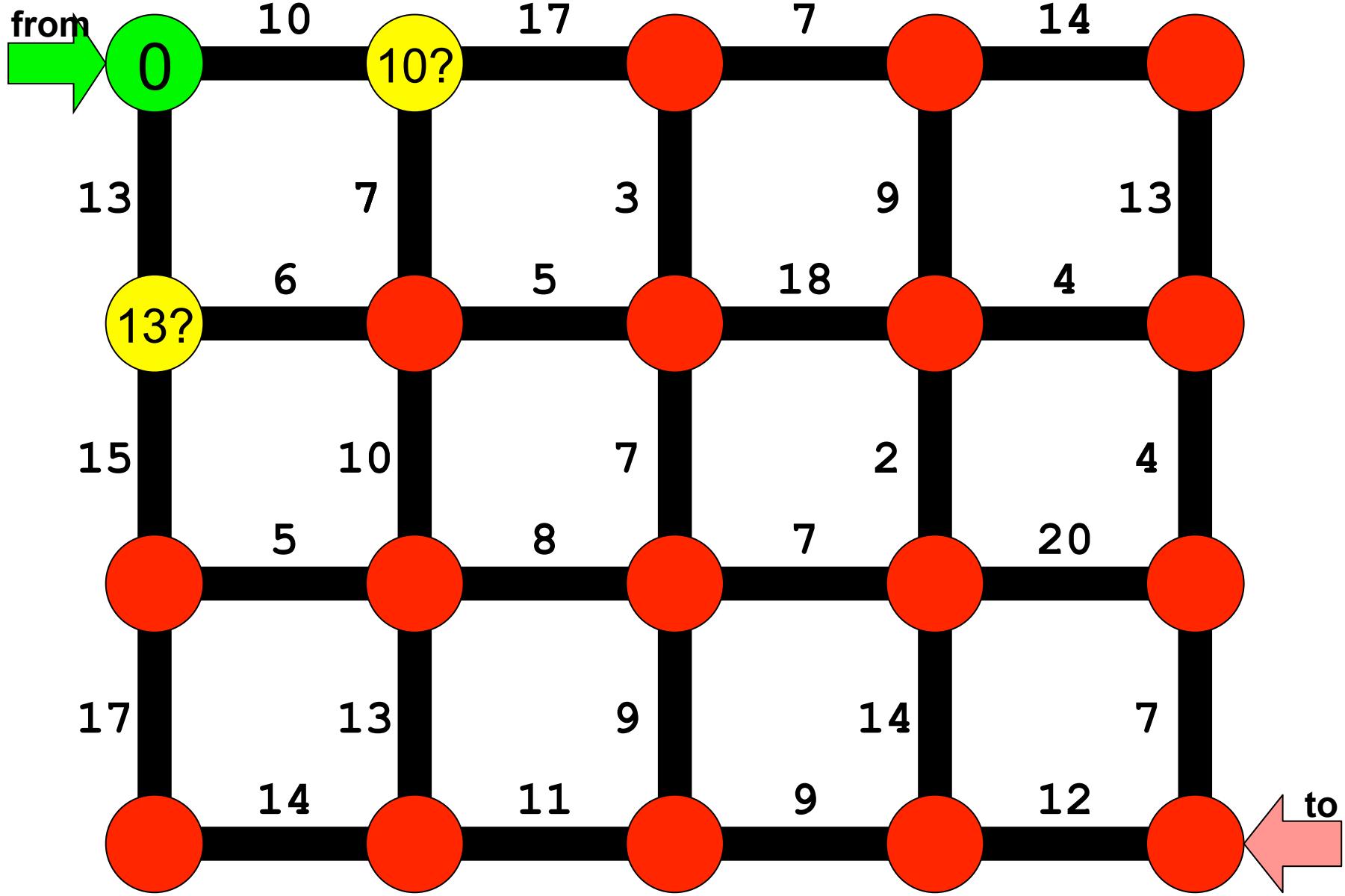
from  
0

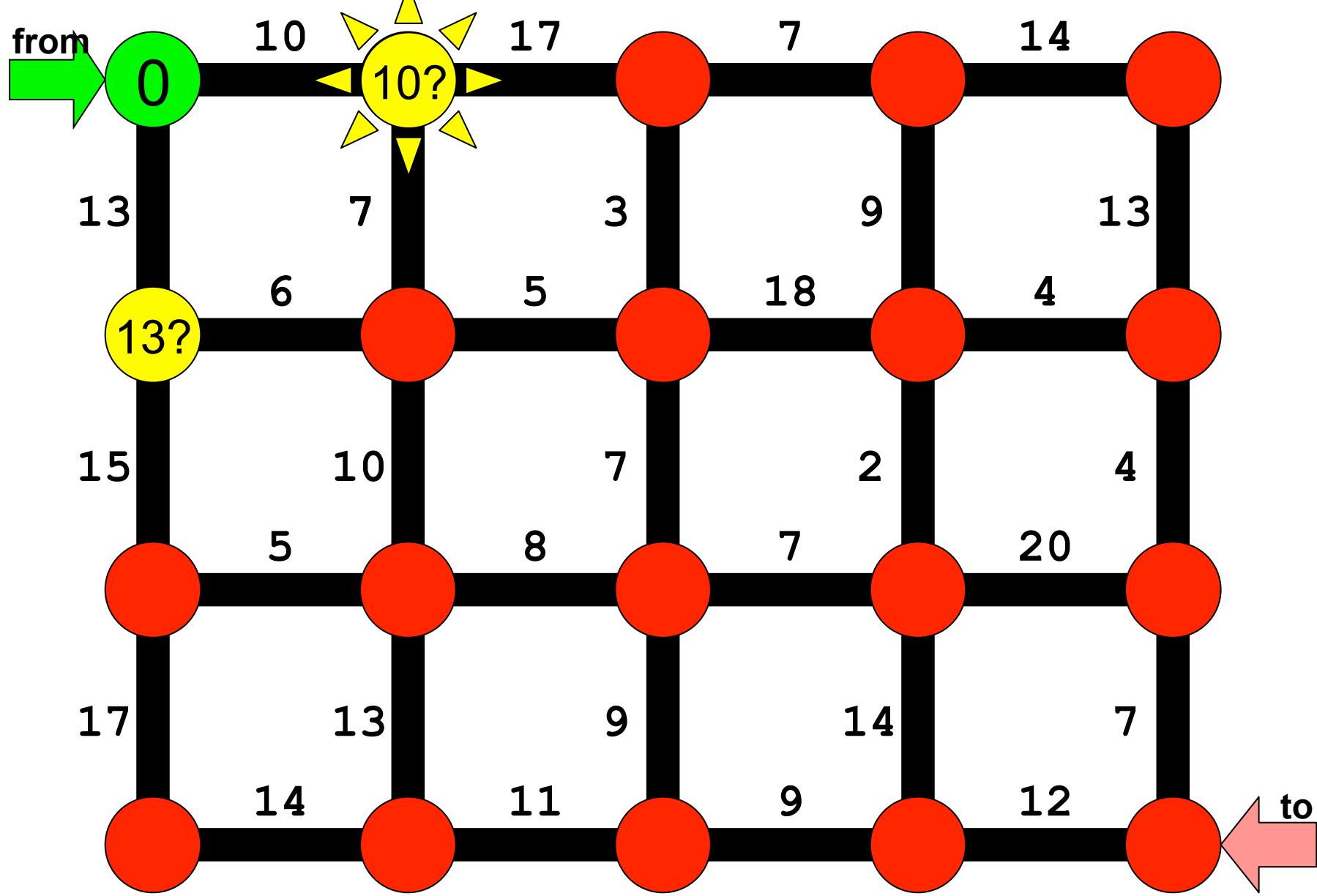


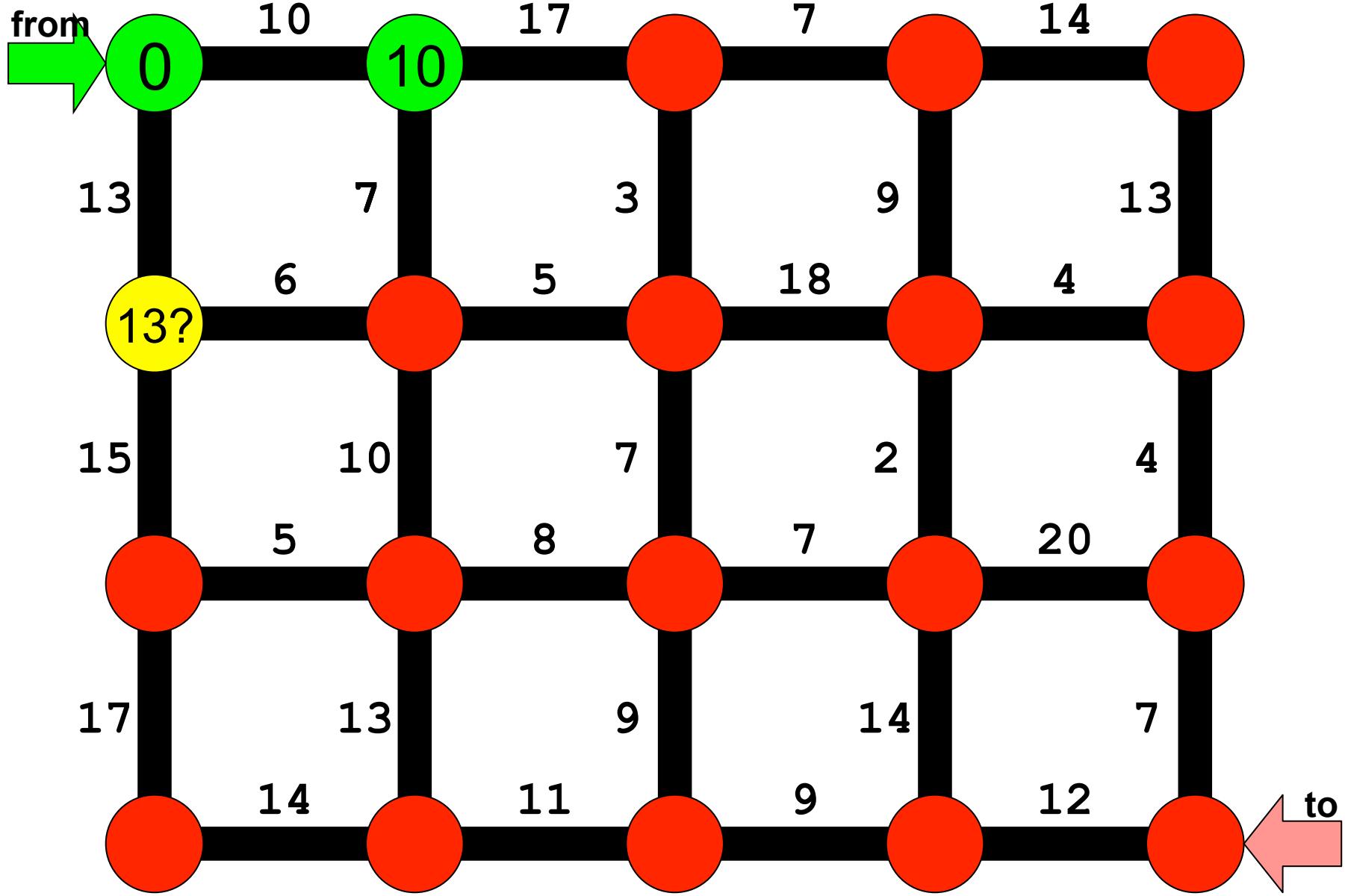
to

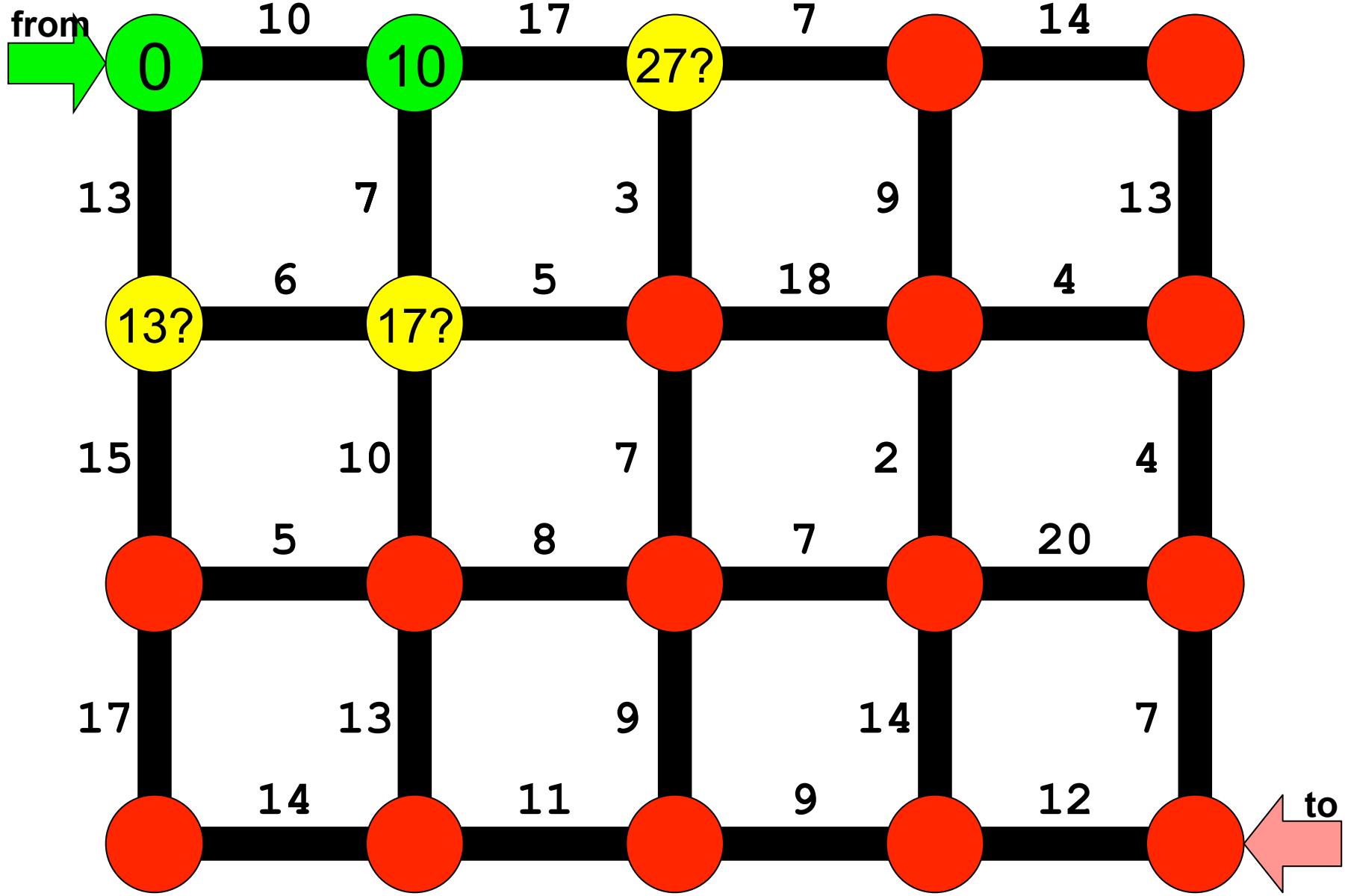
from



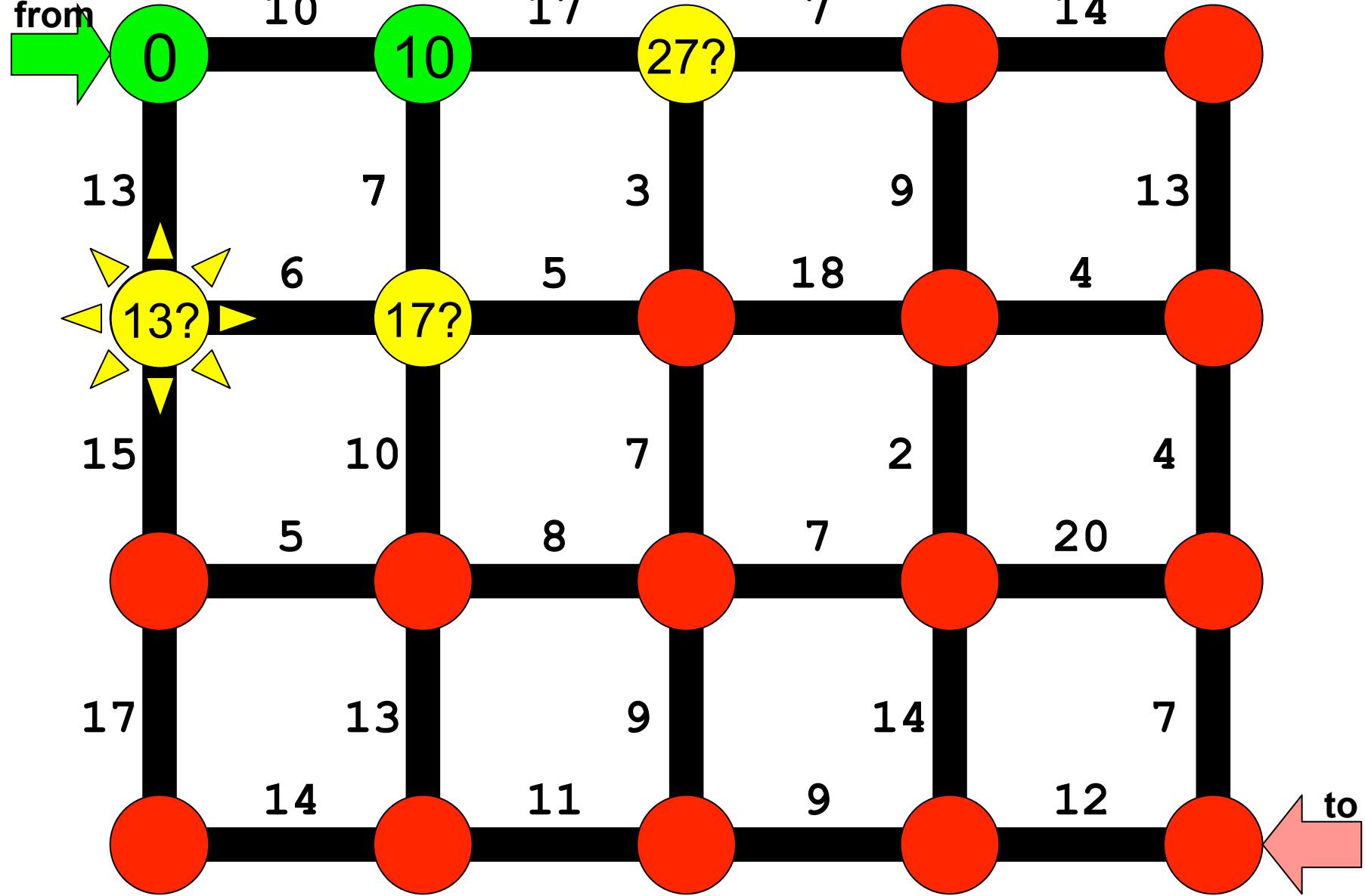




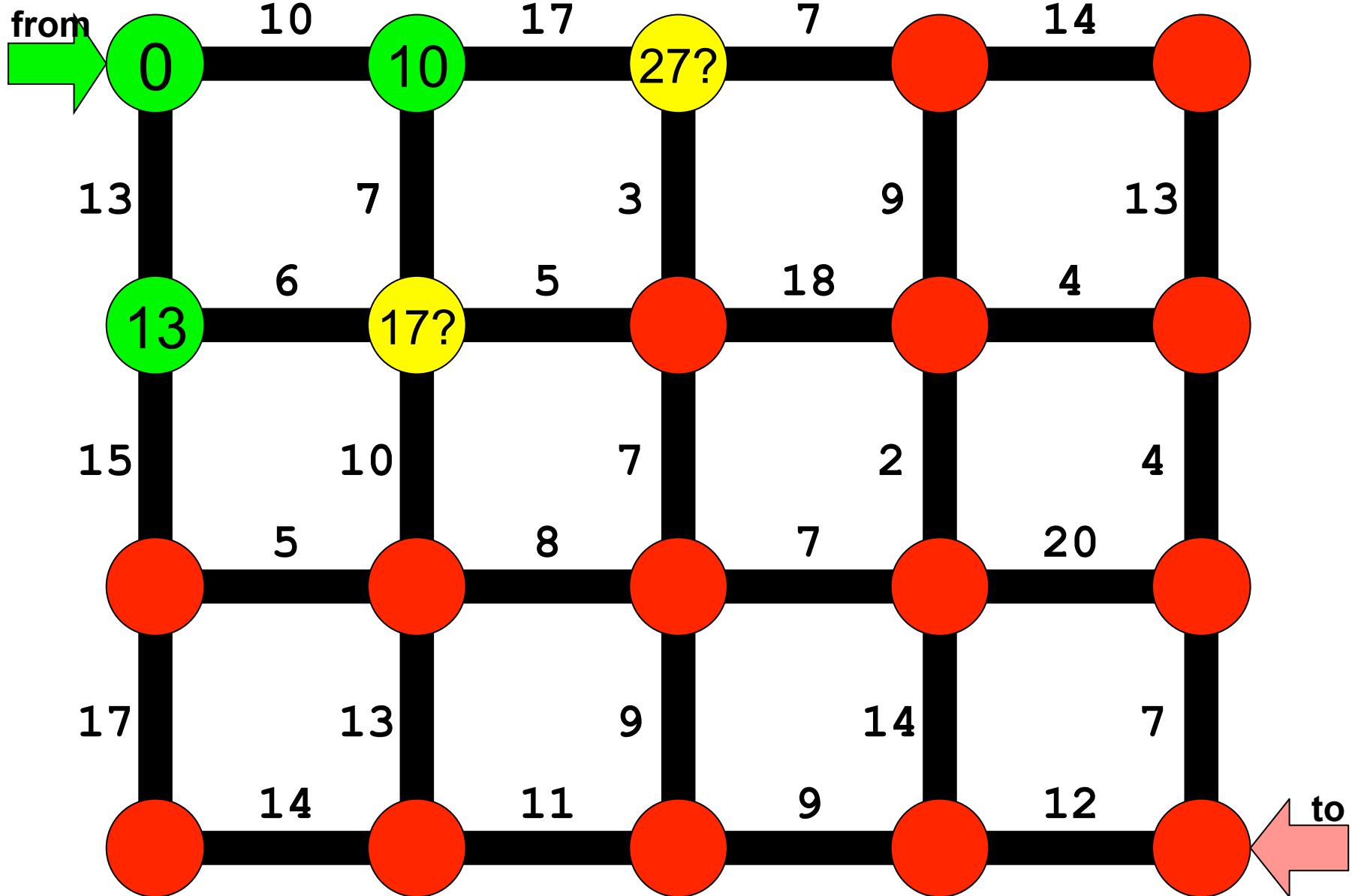


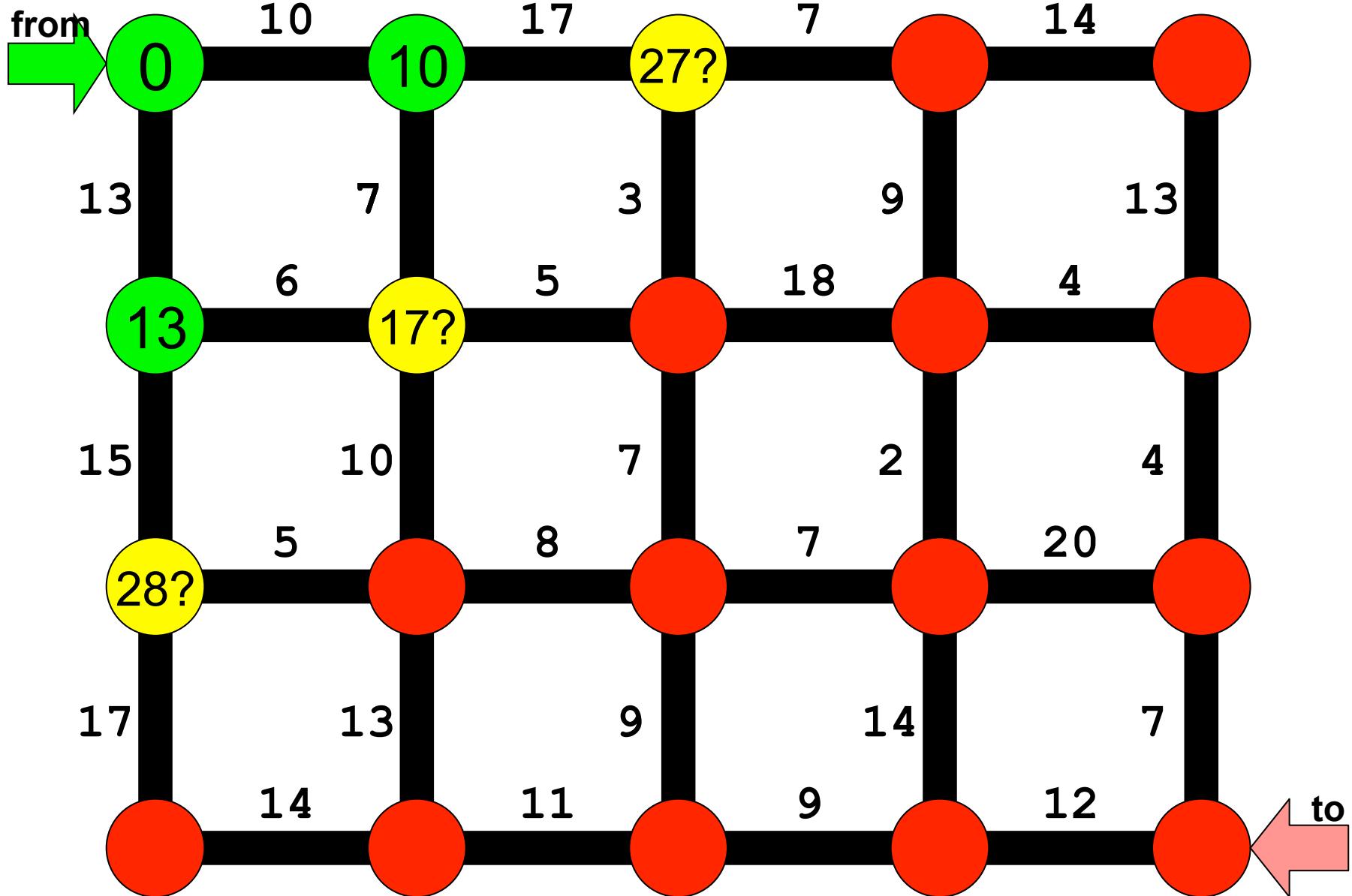


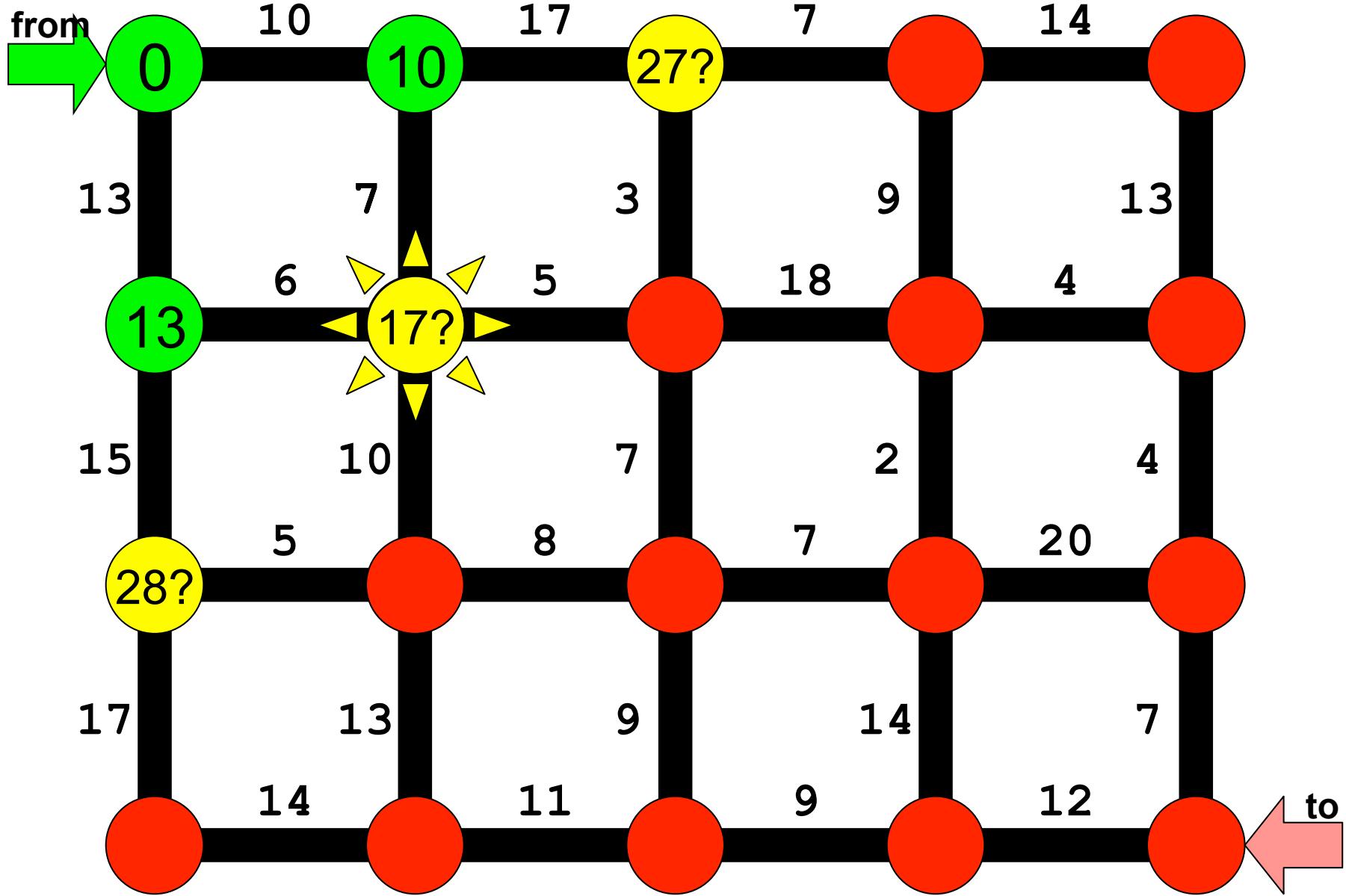
from

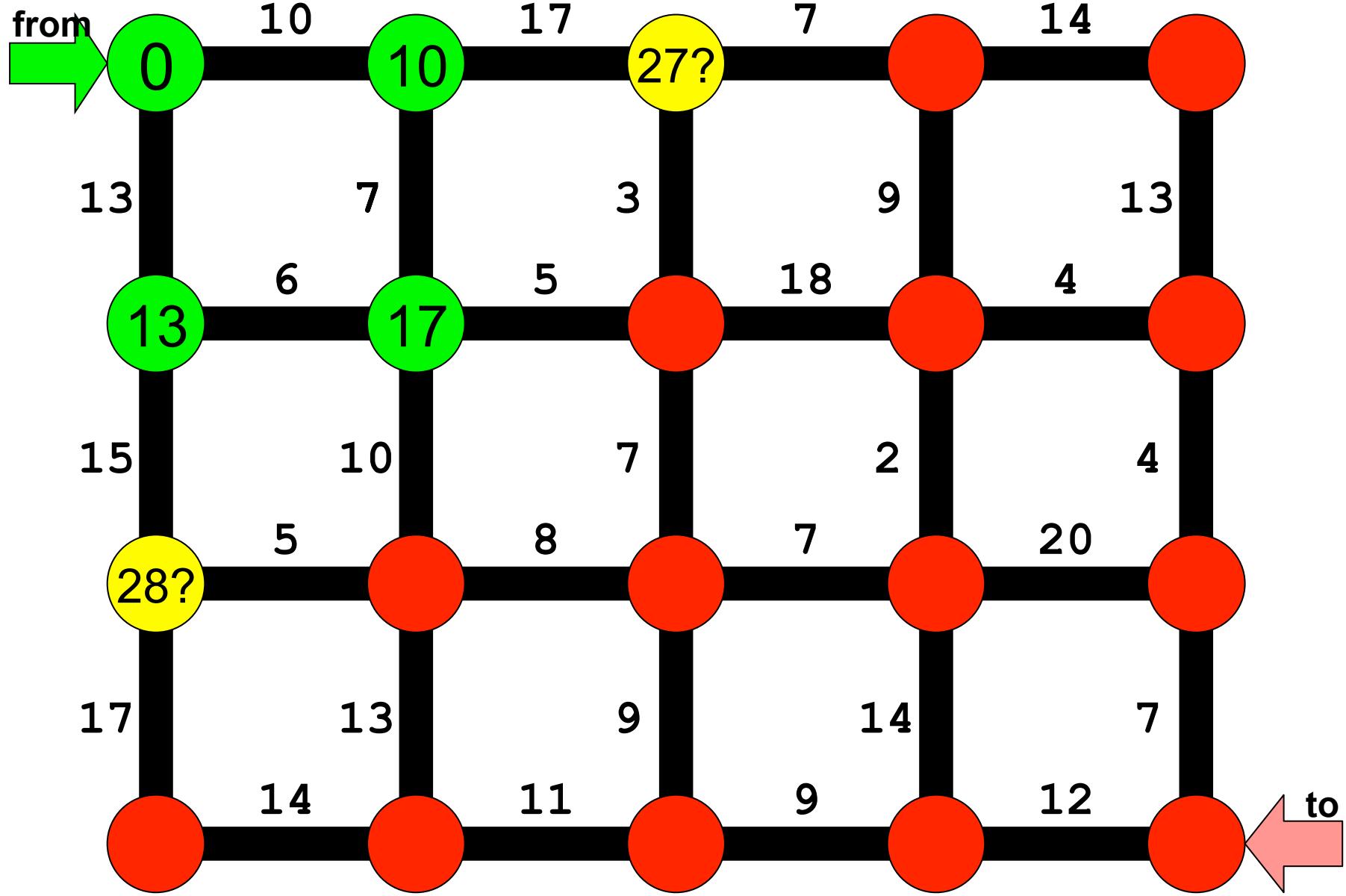


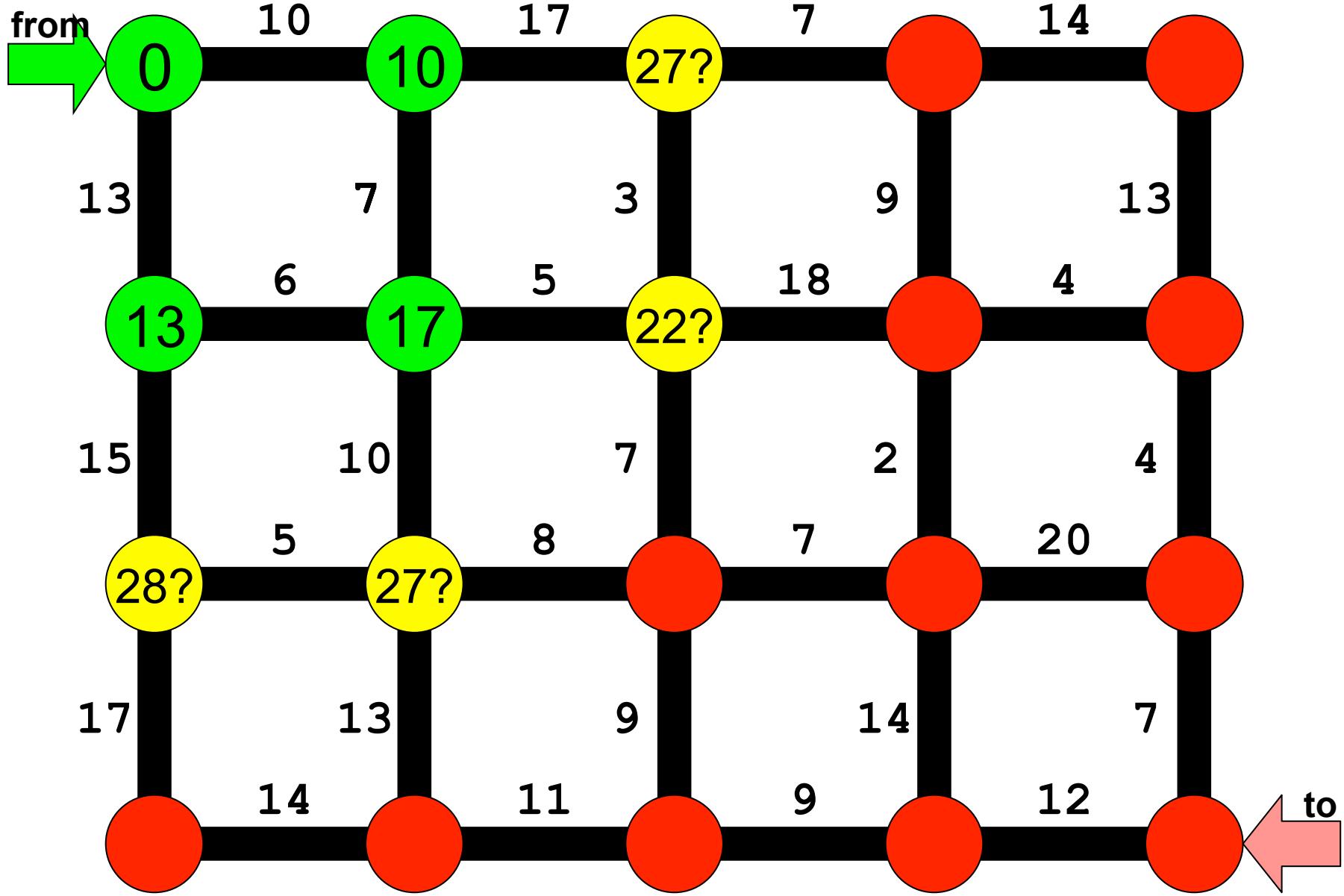
to

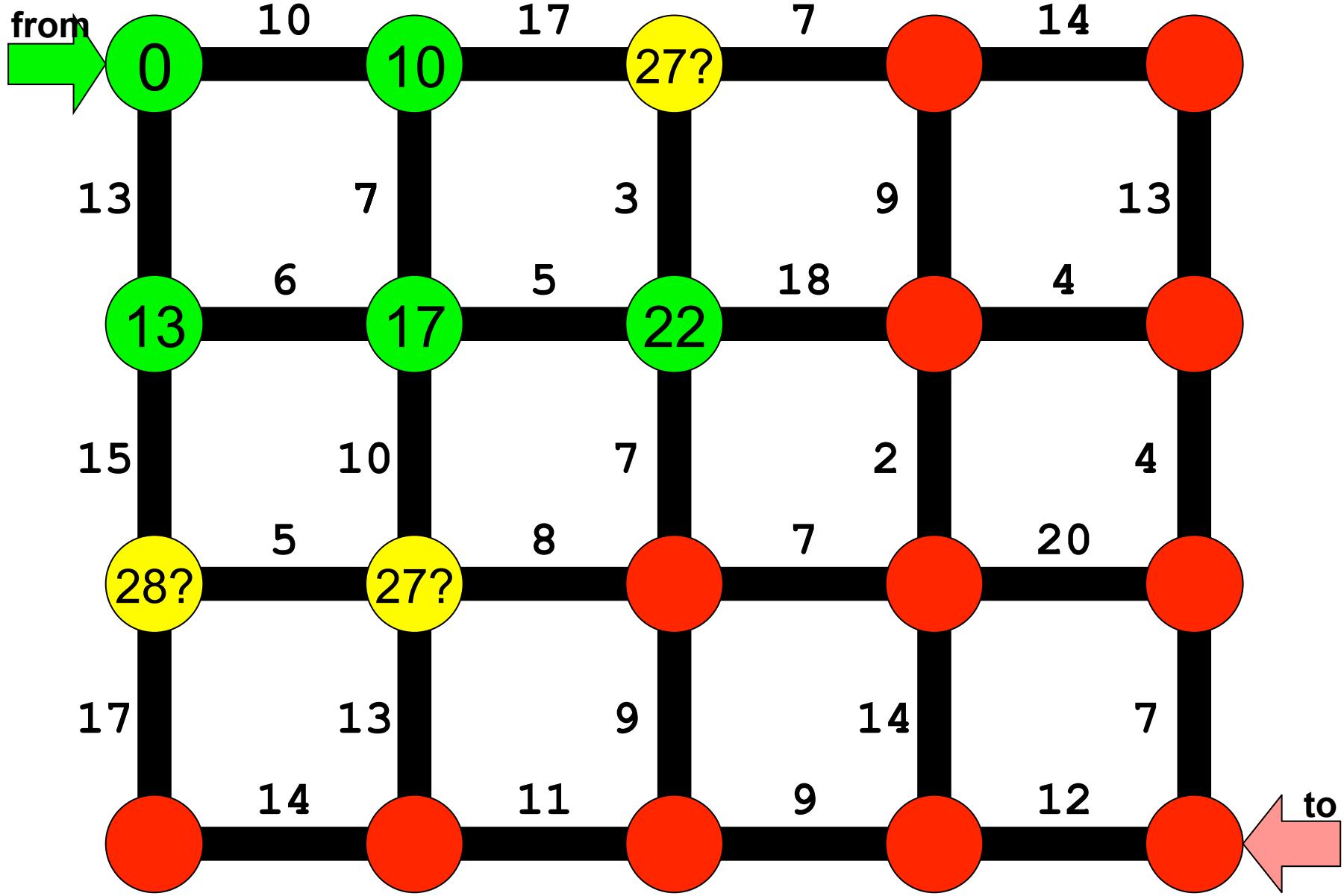


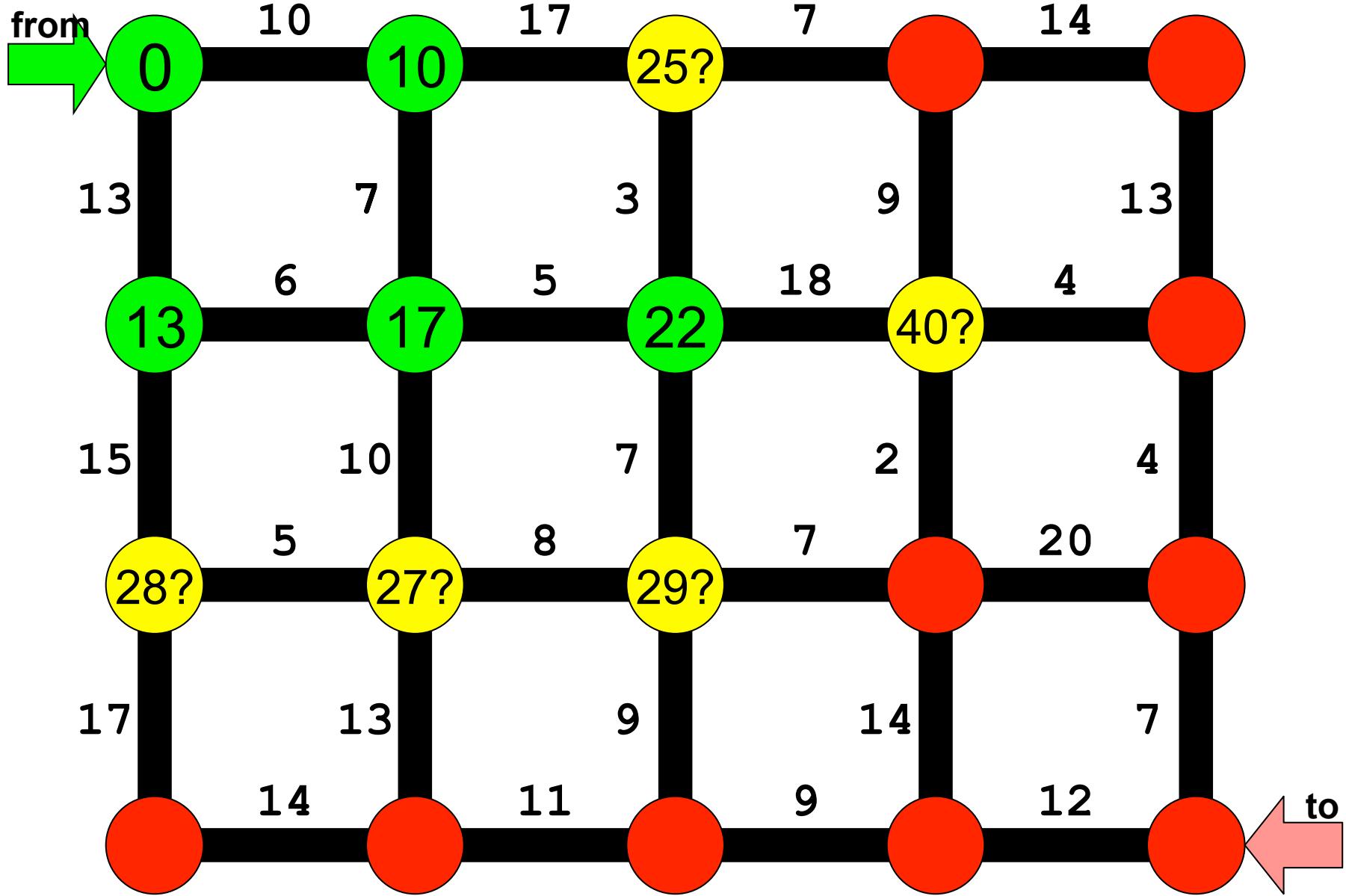


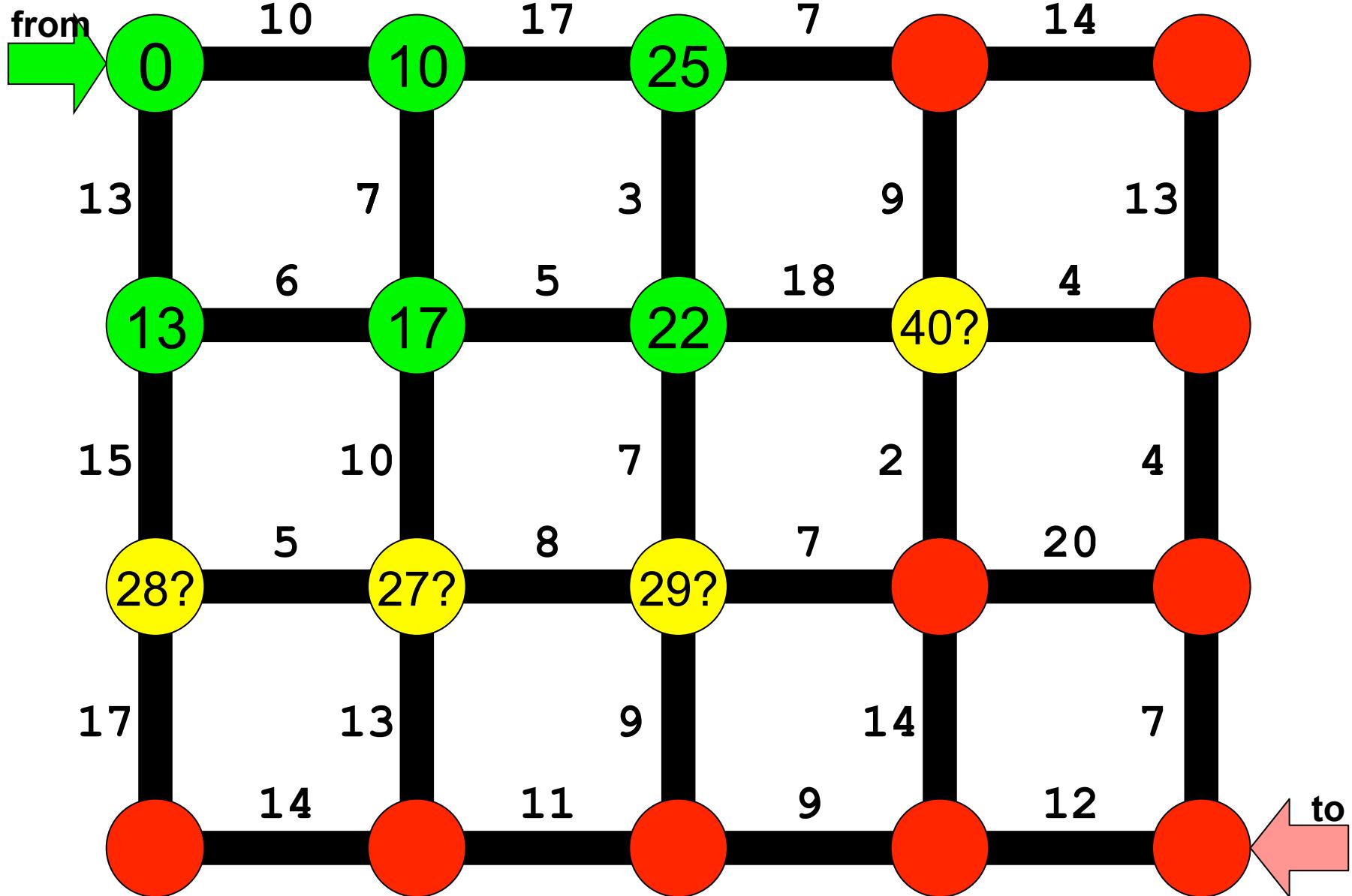


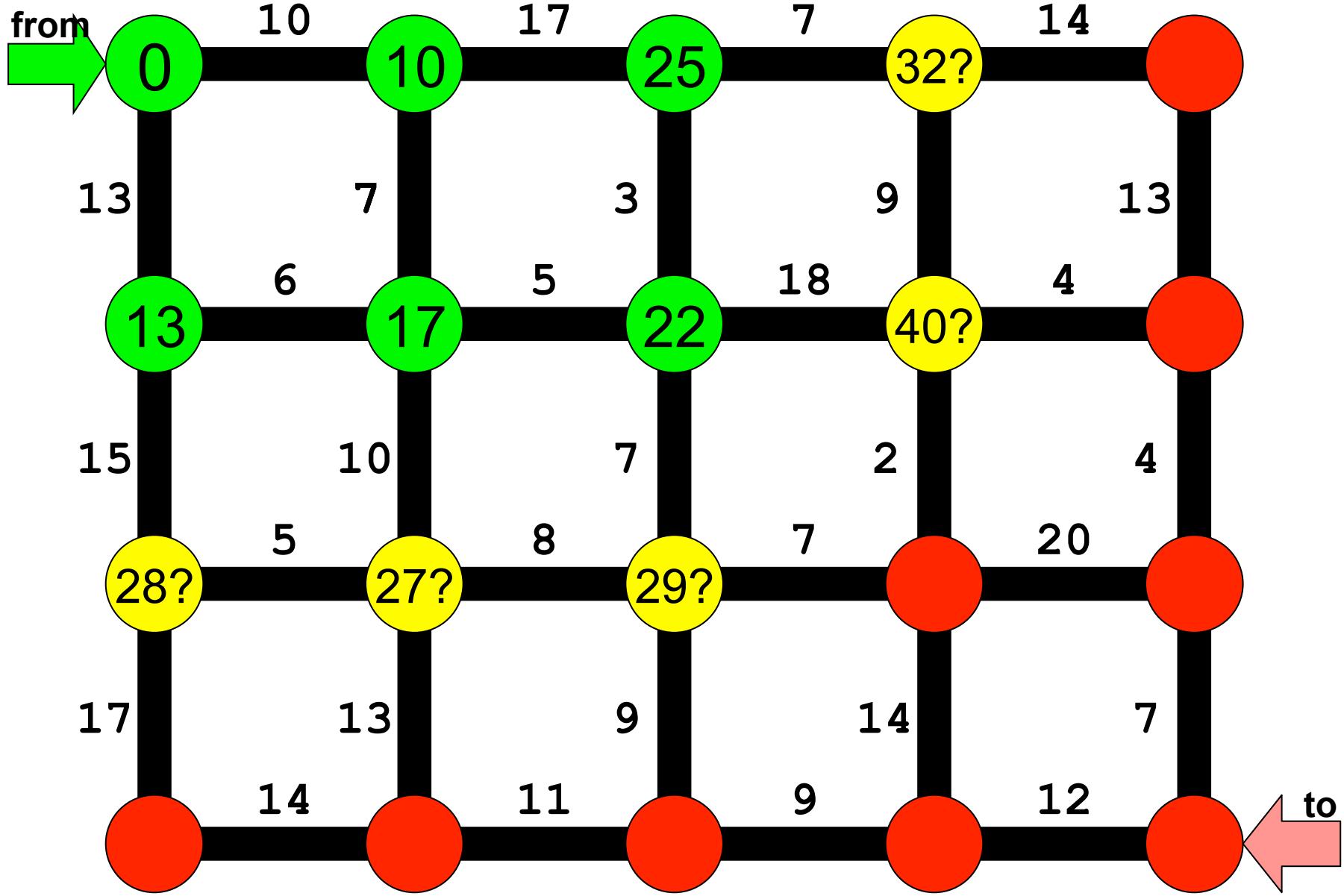


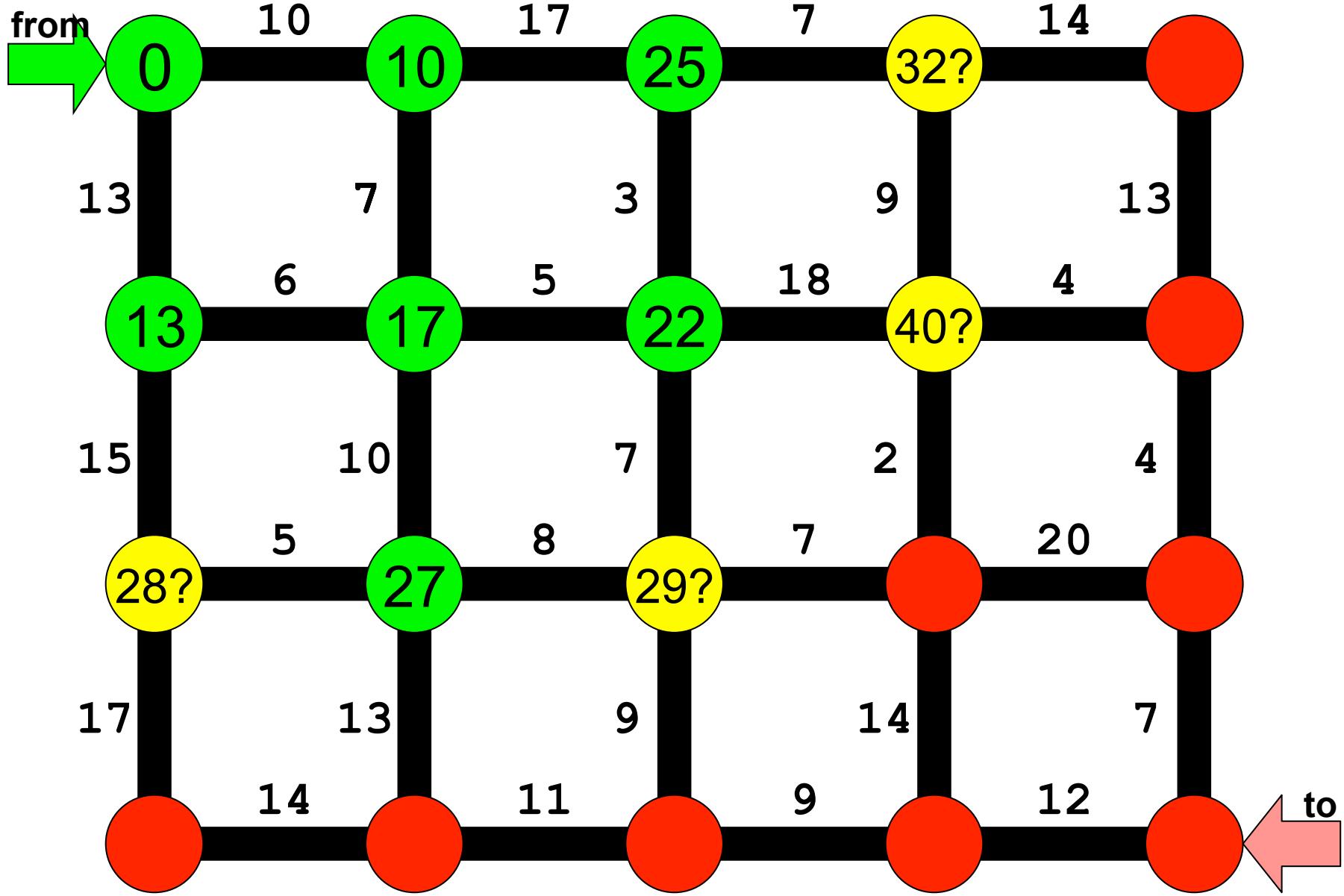


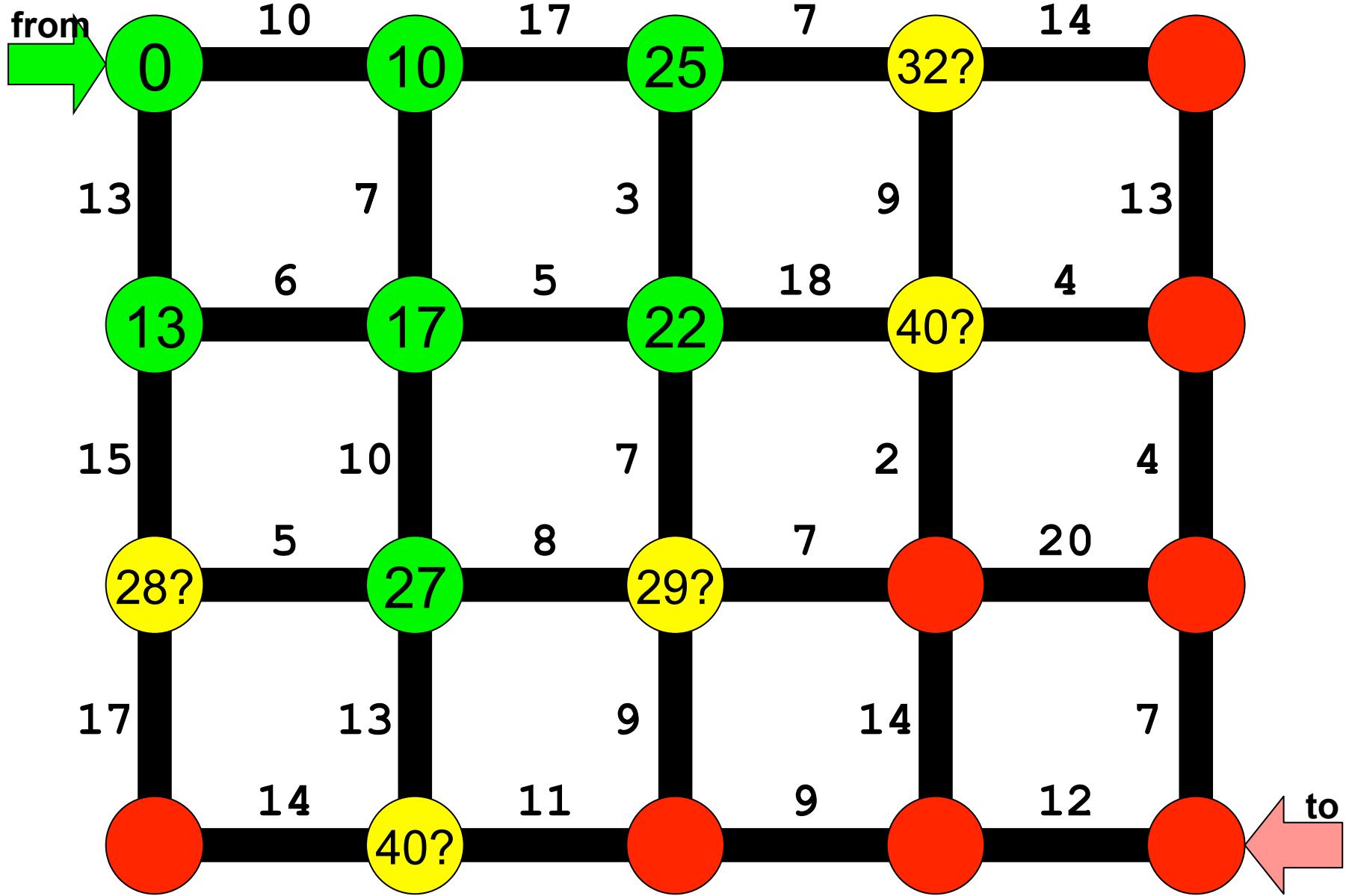


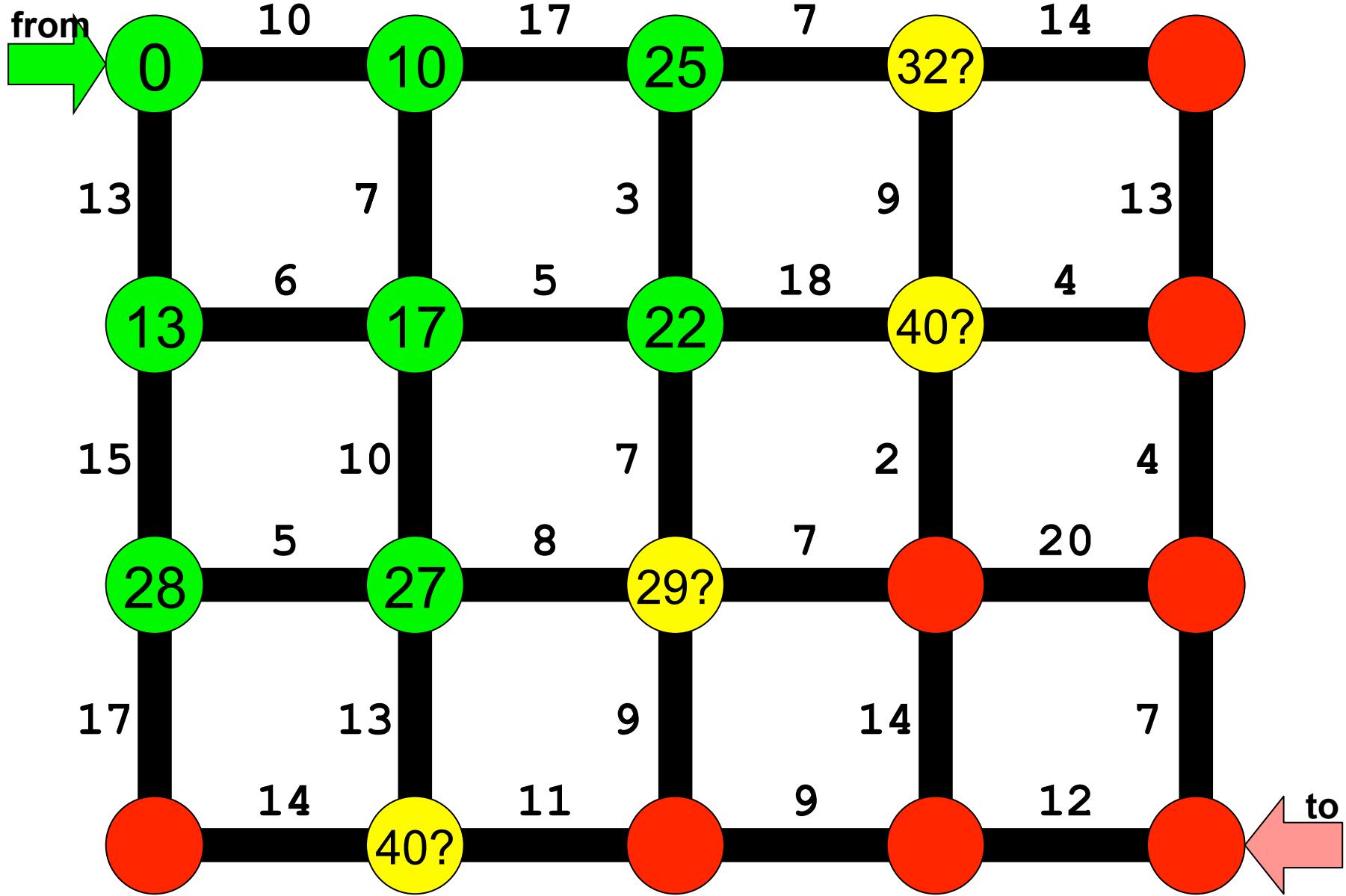


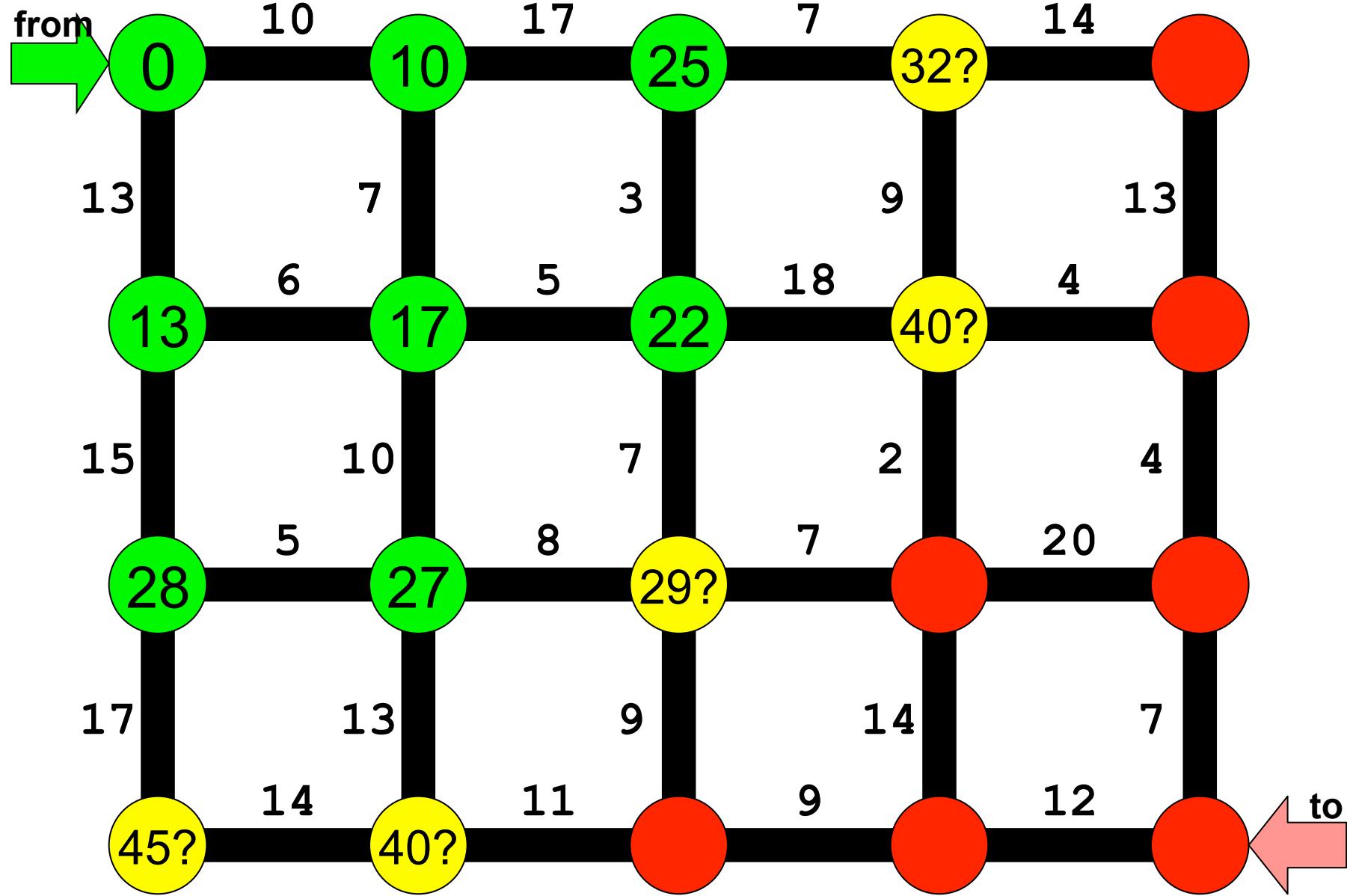


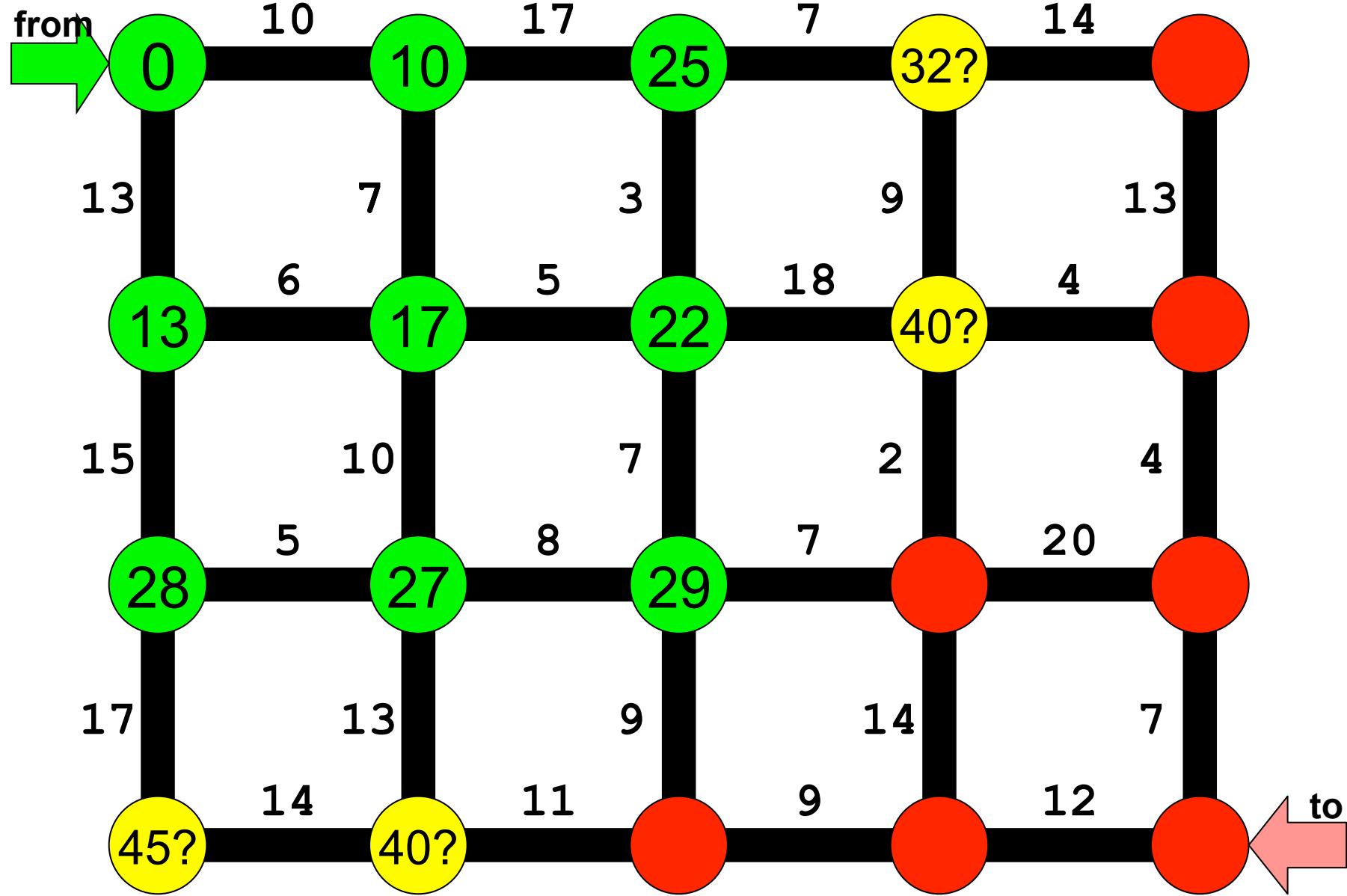


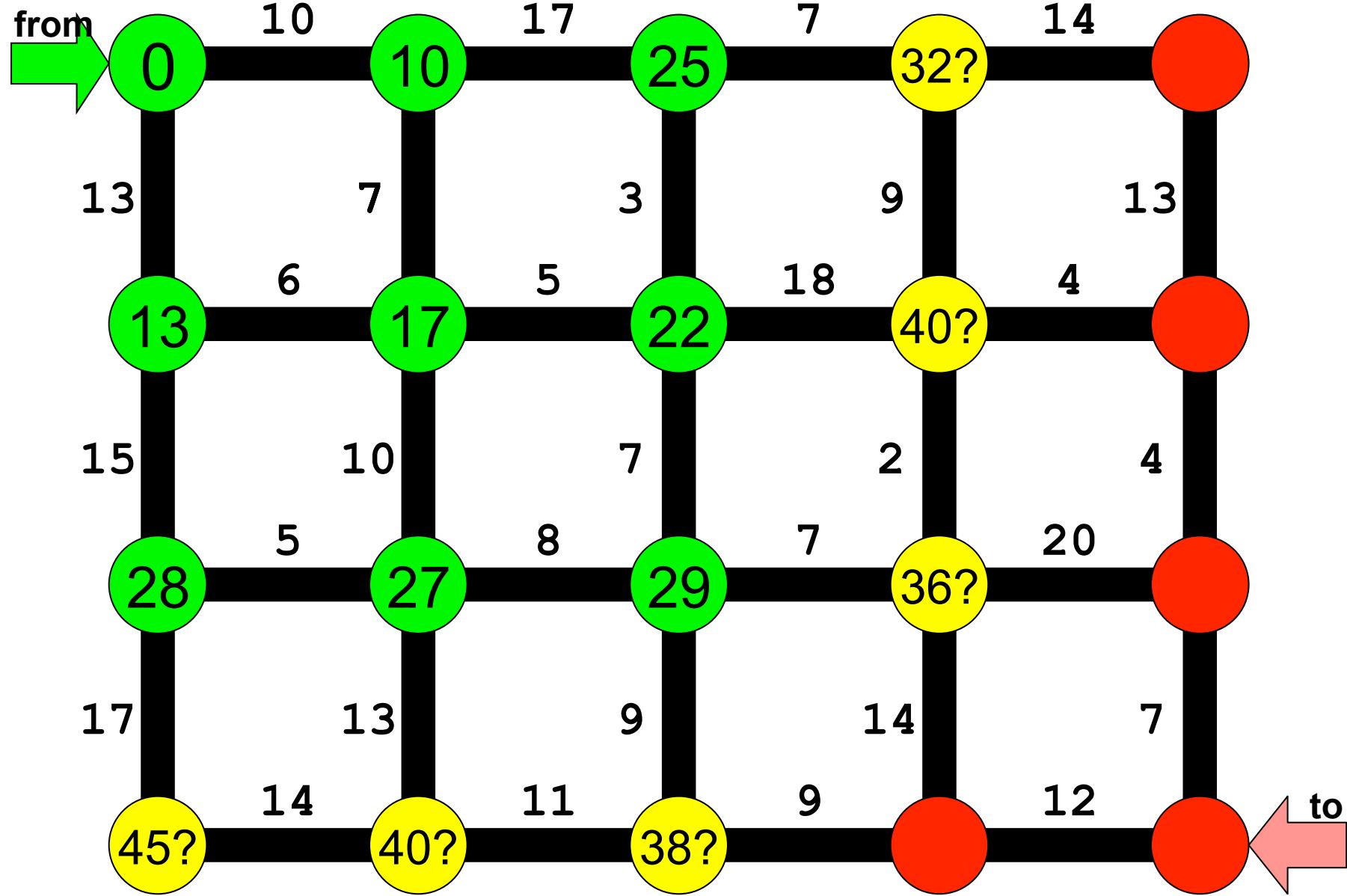


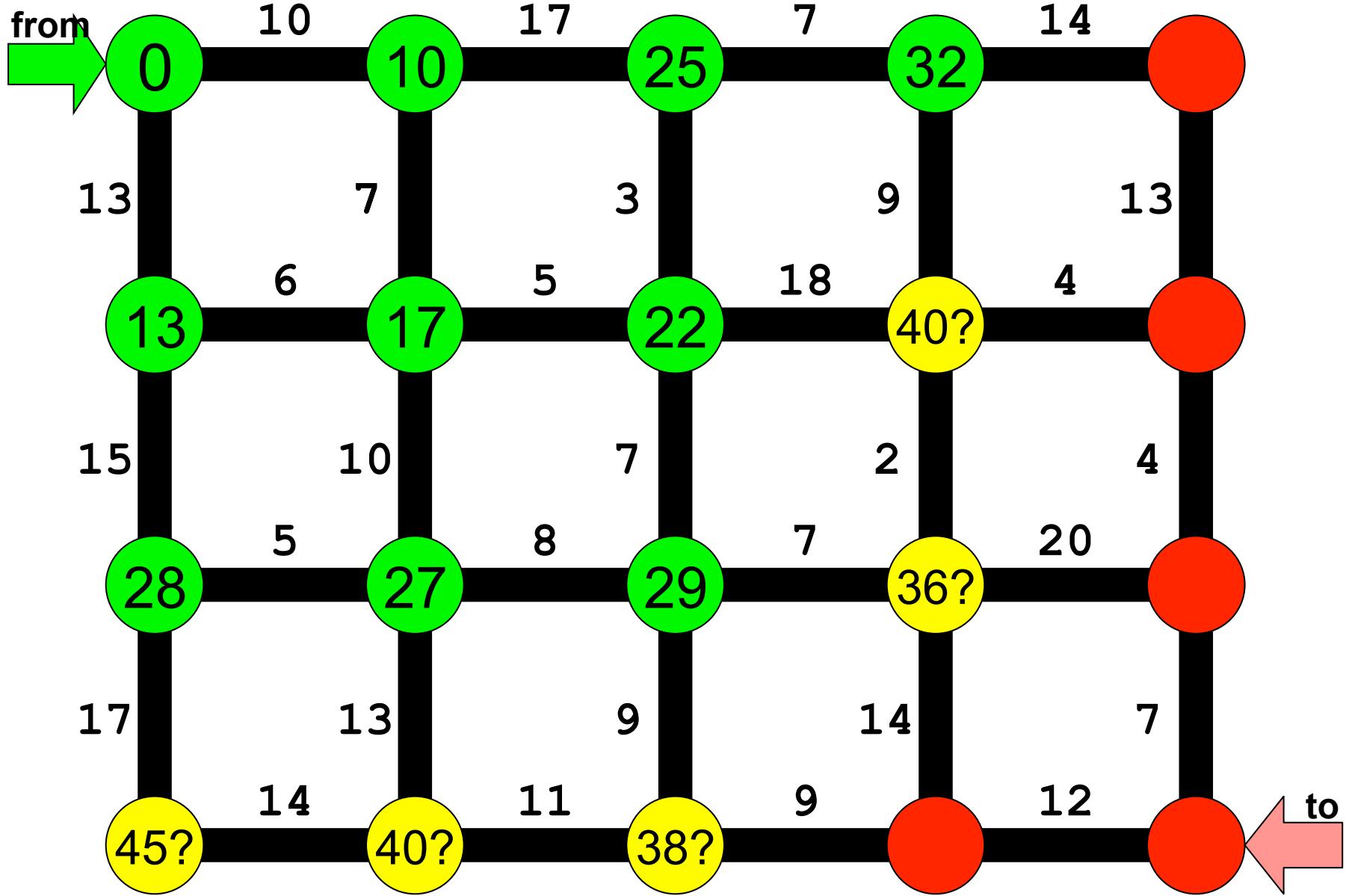


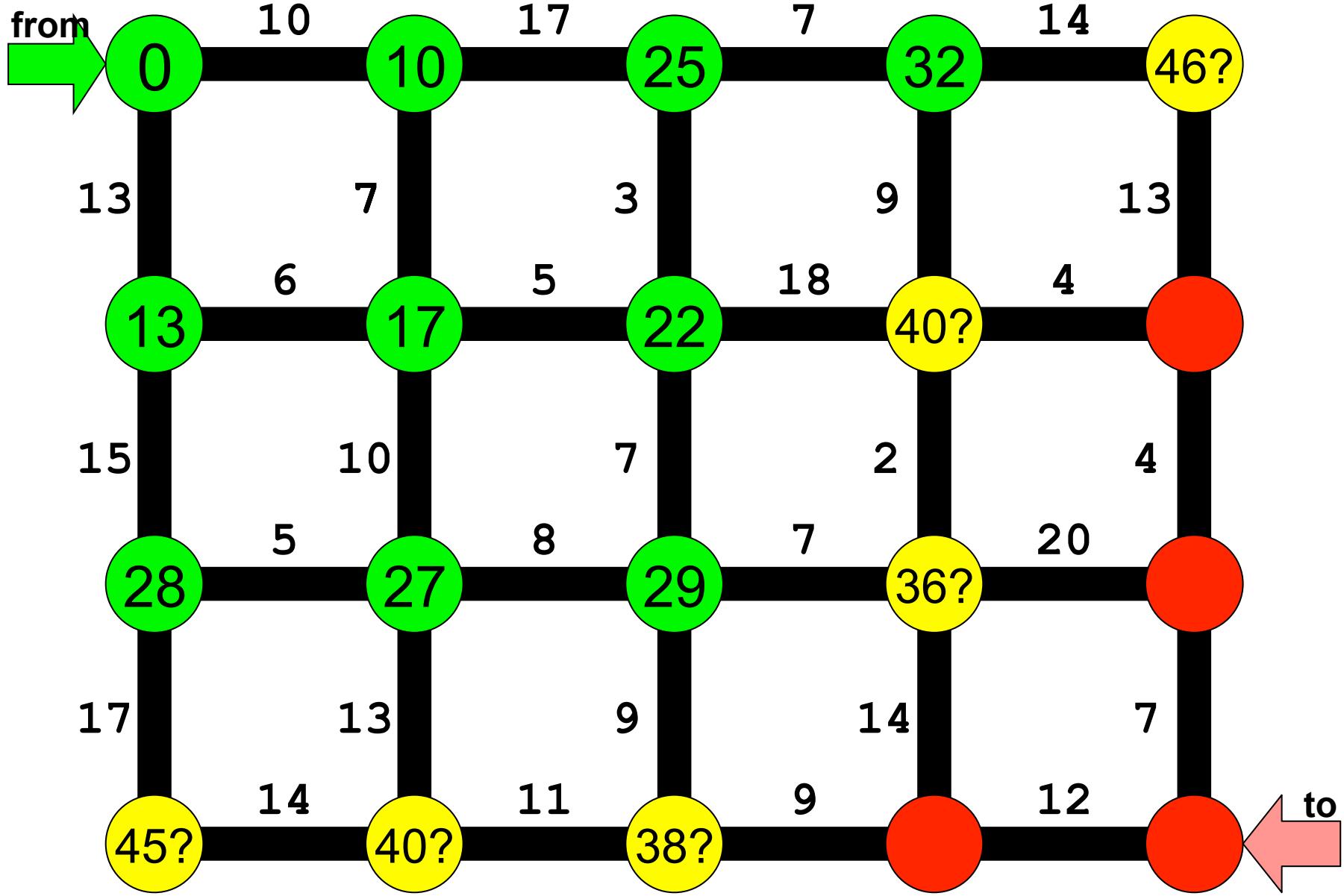


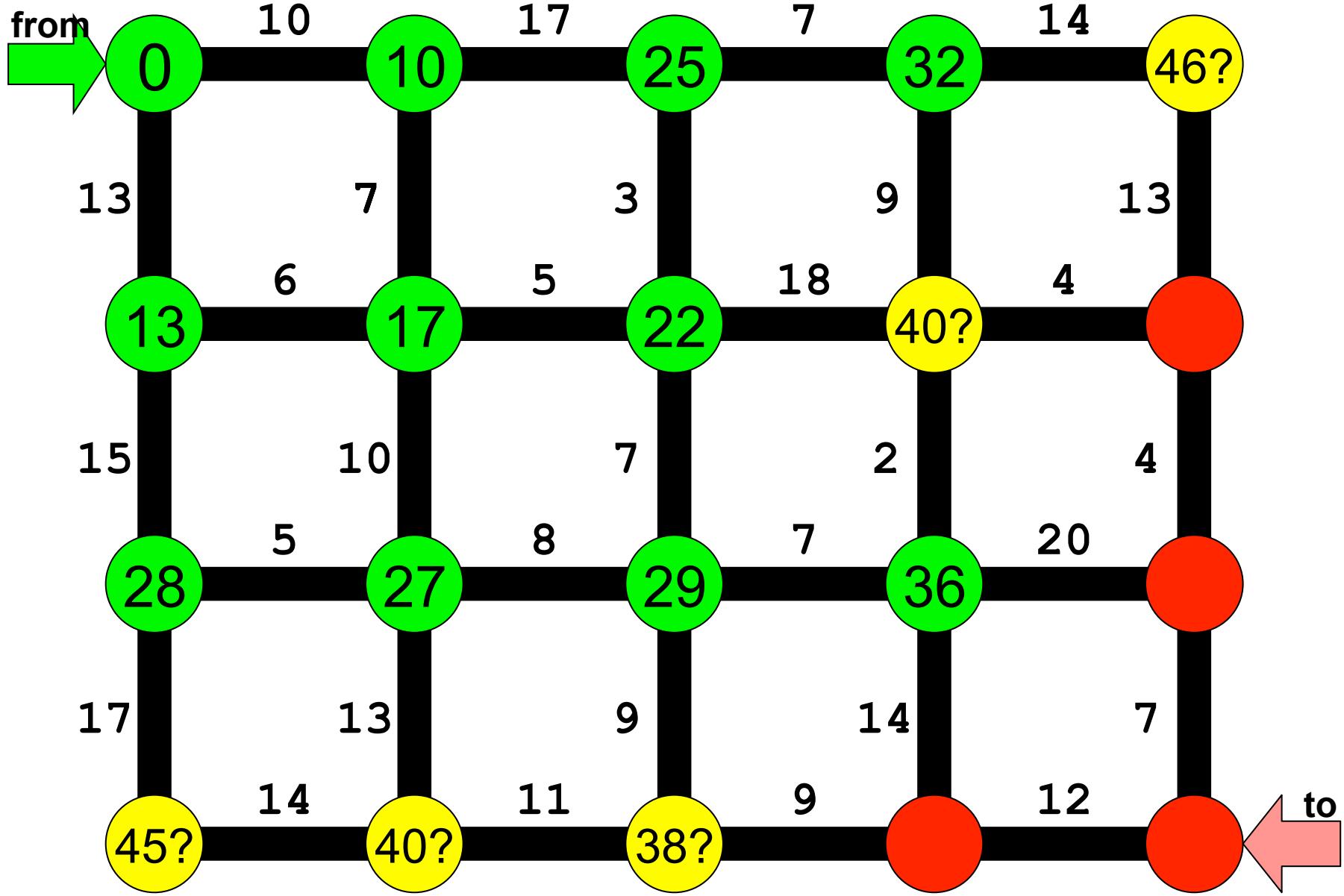


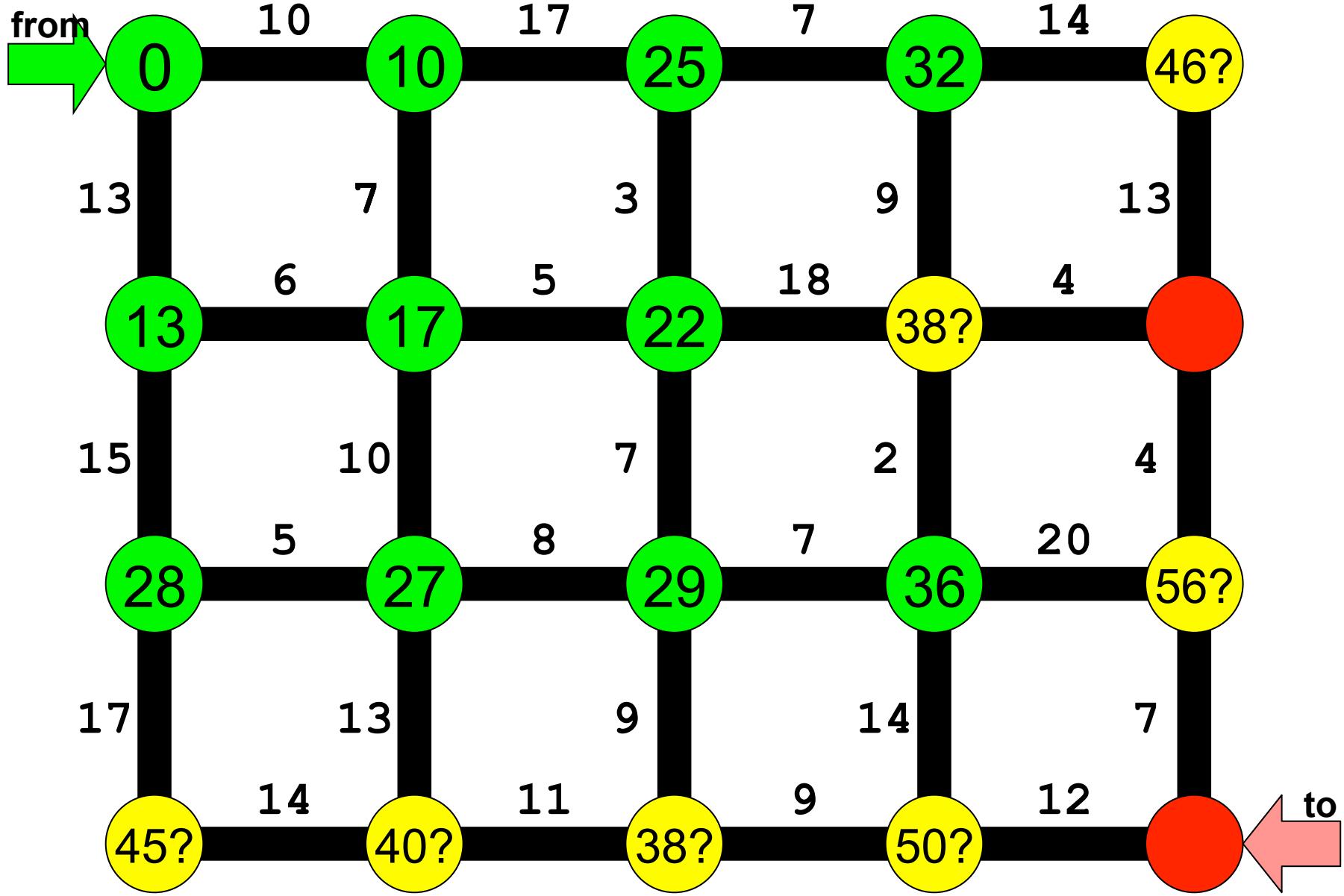


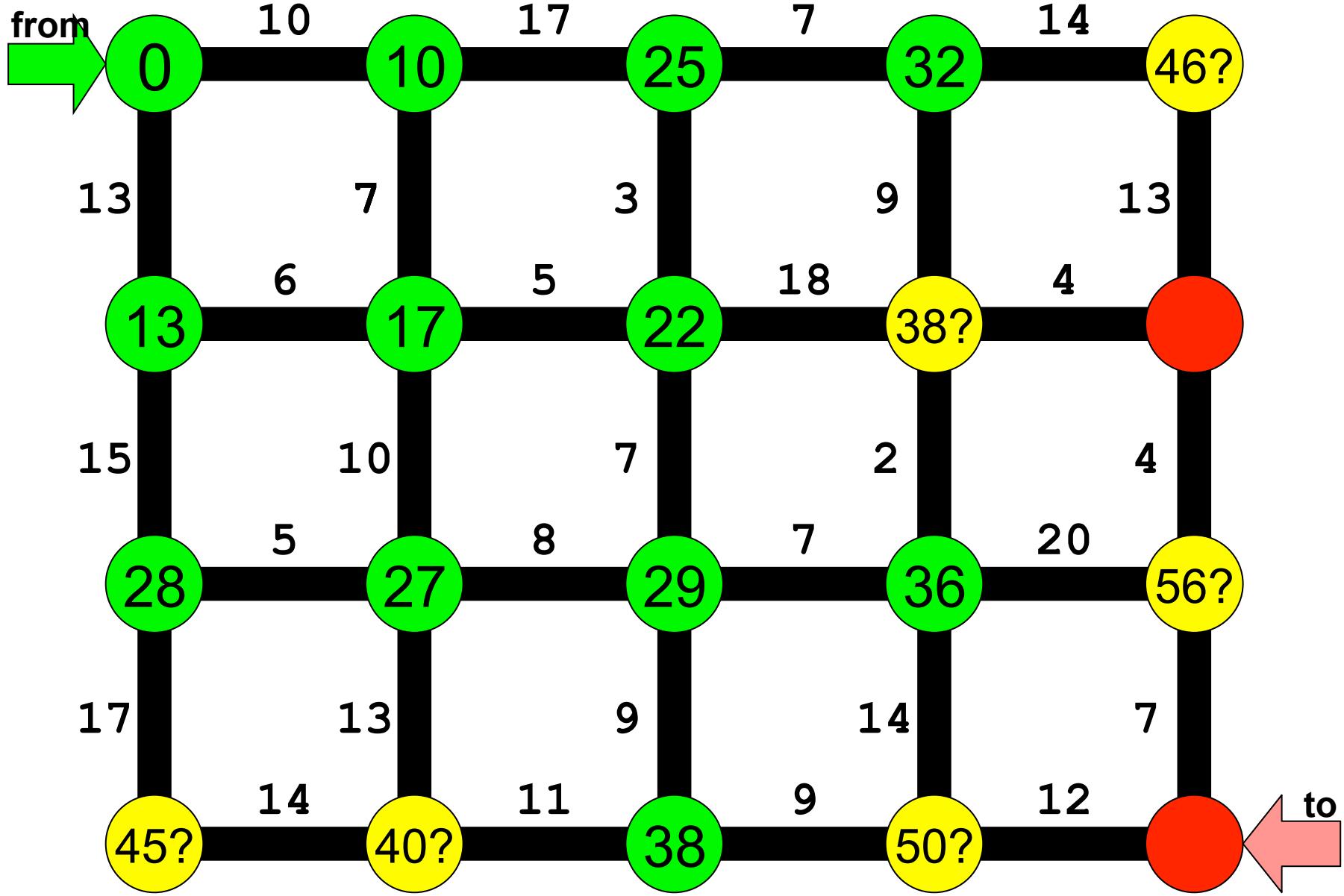


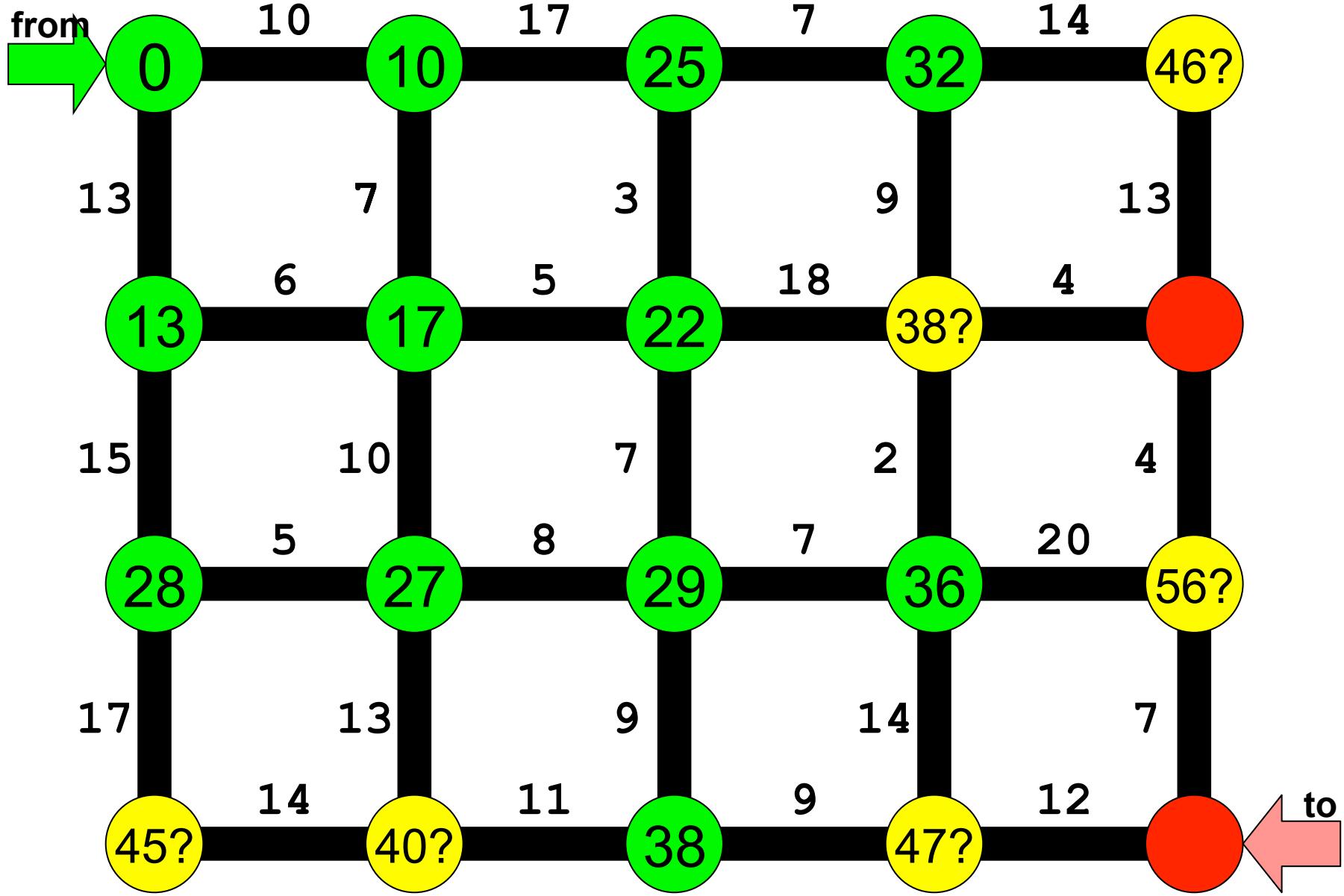


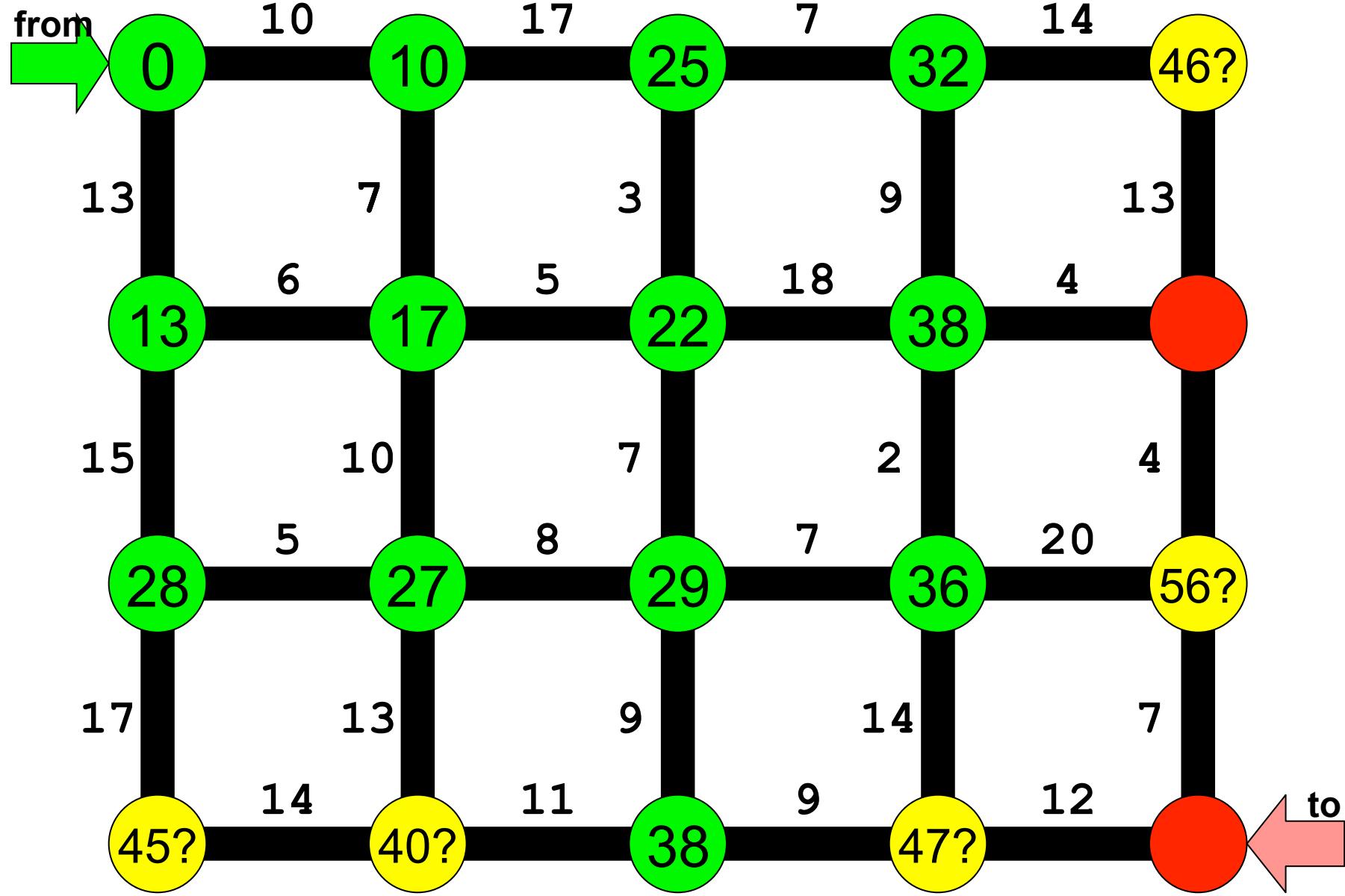


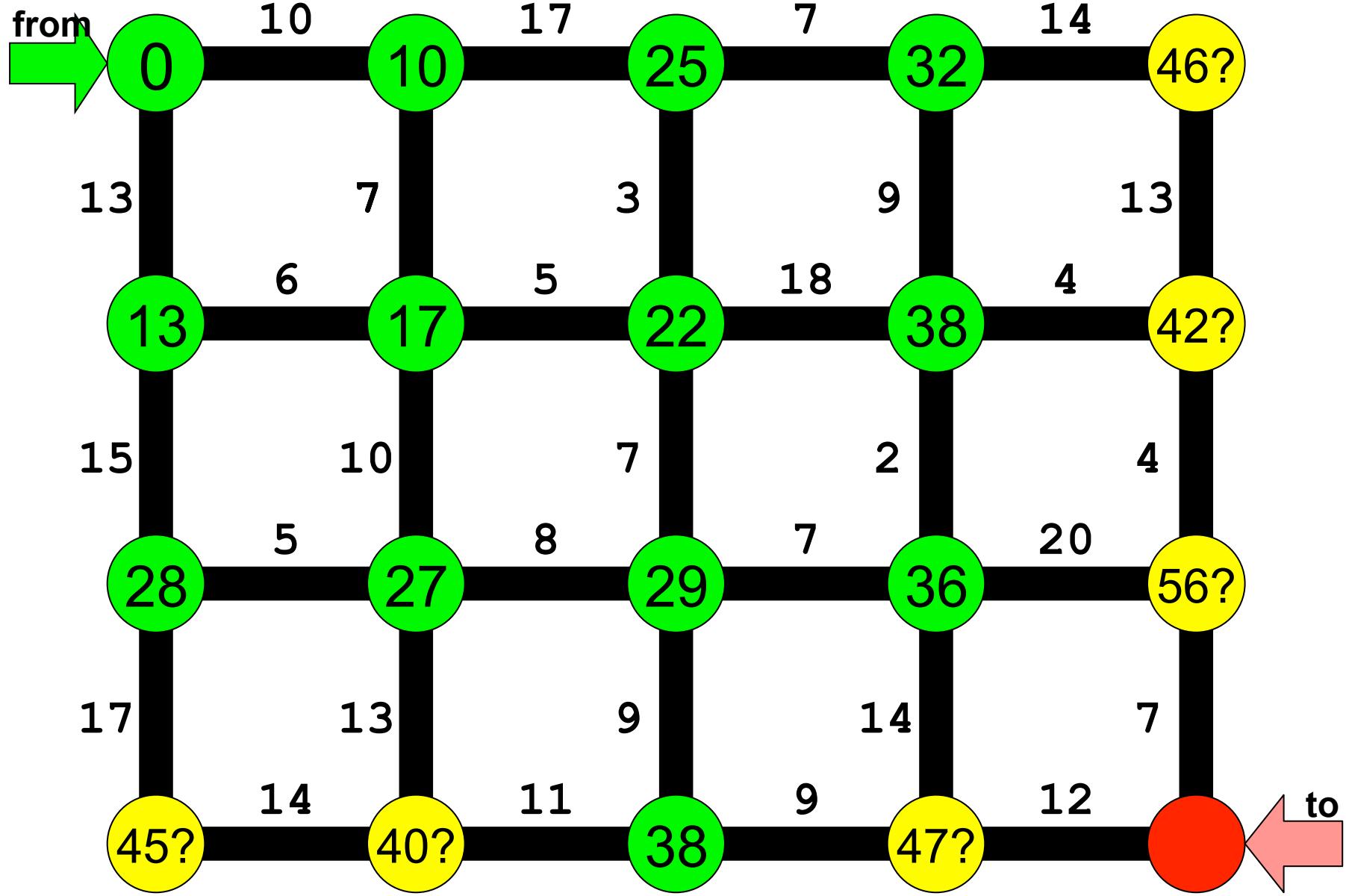


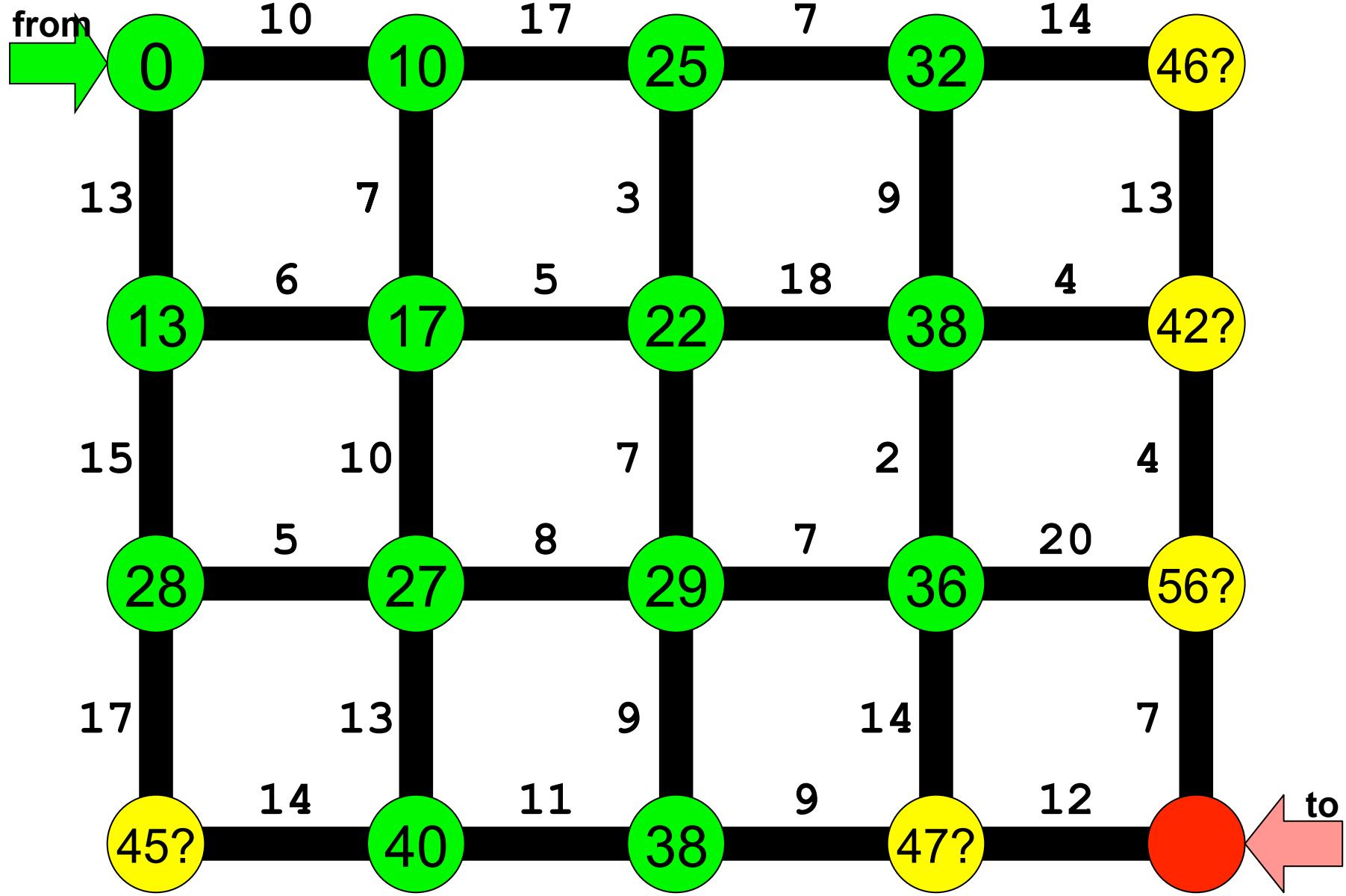


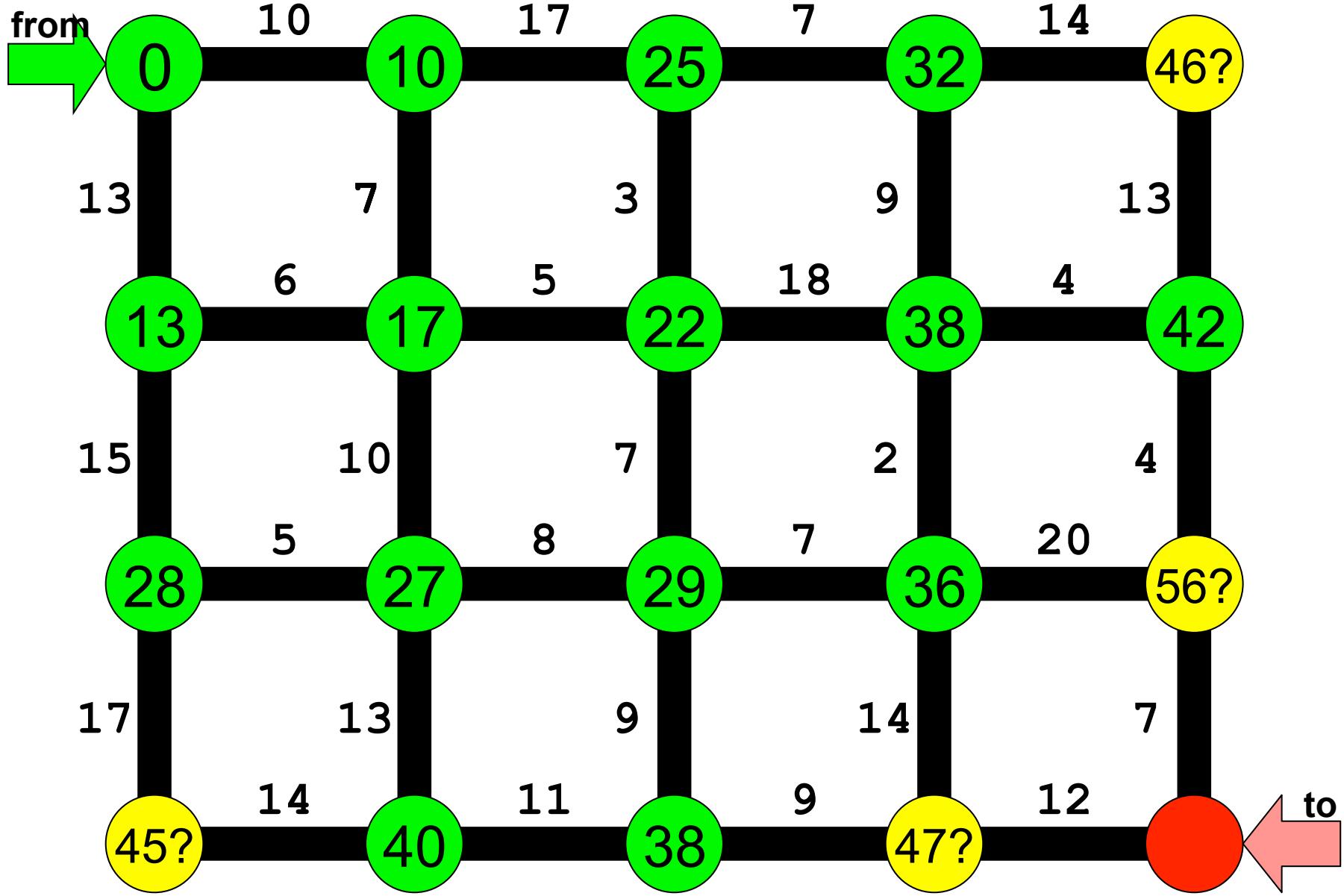


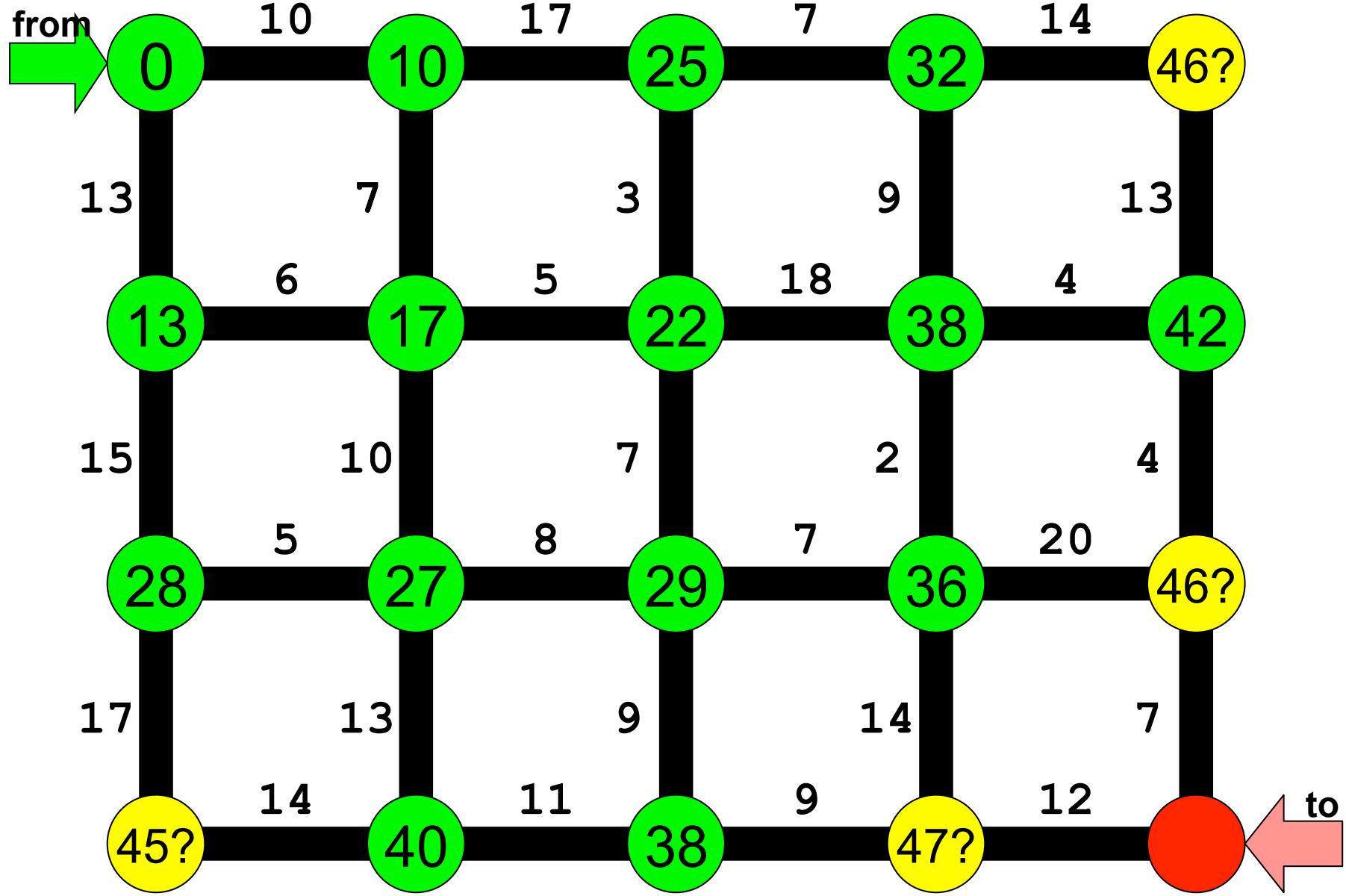


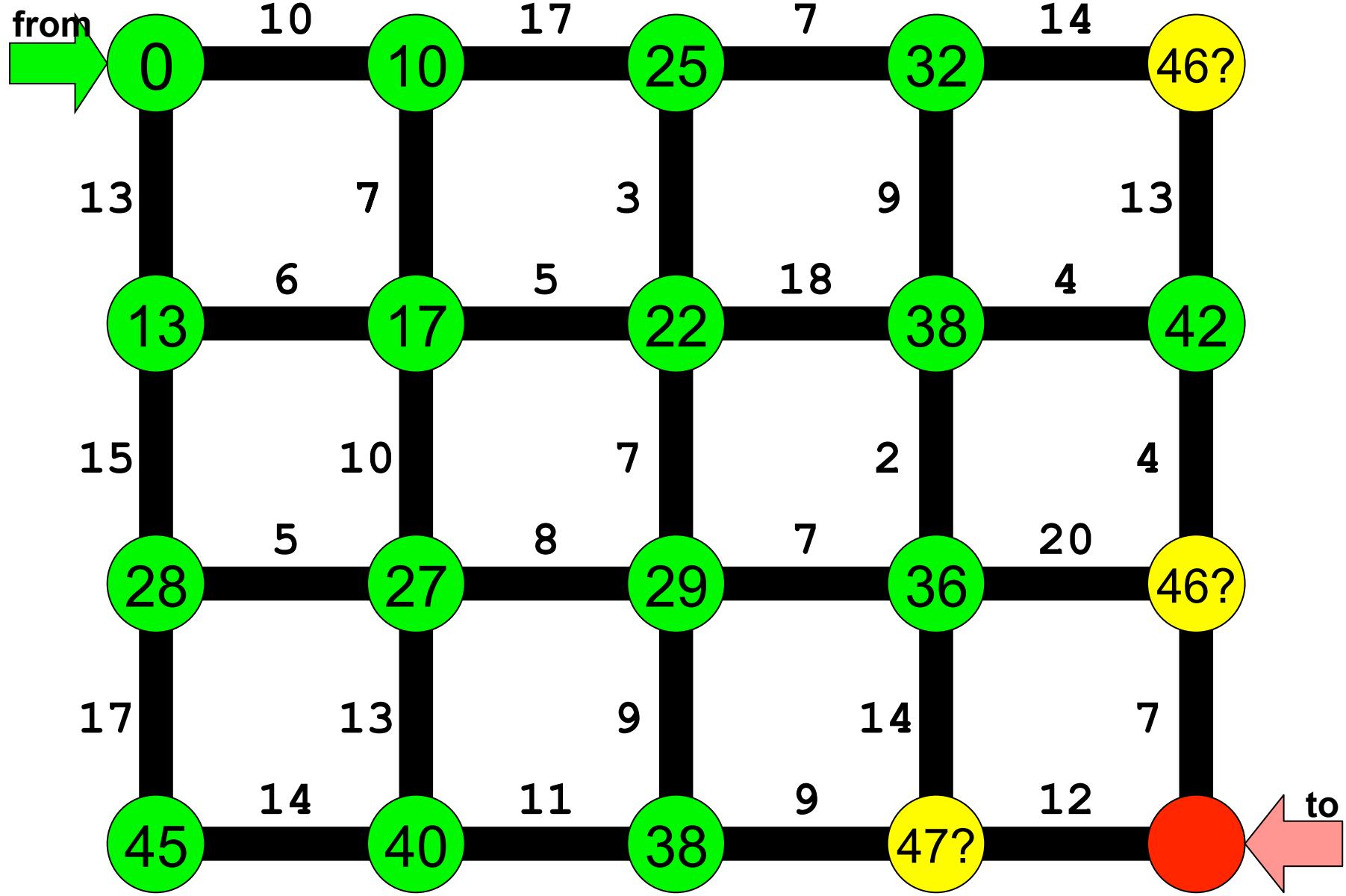


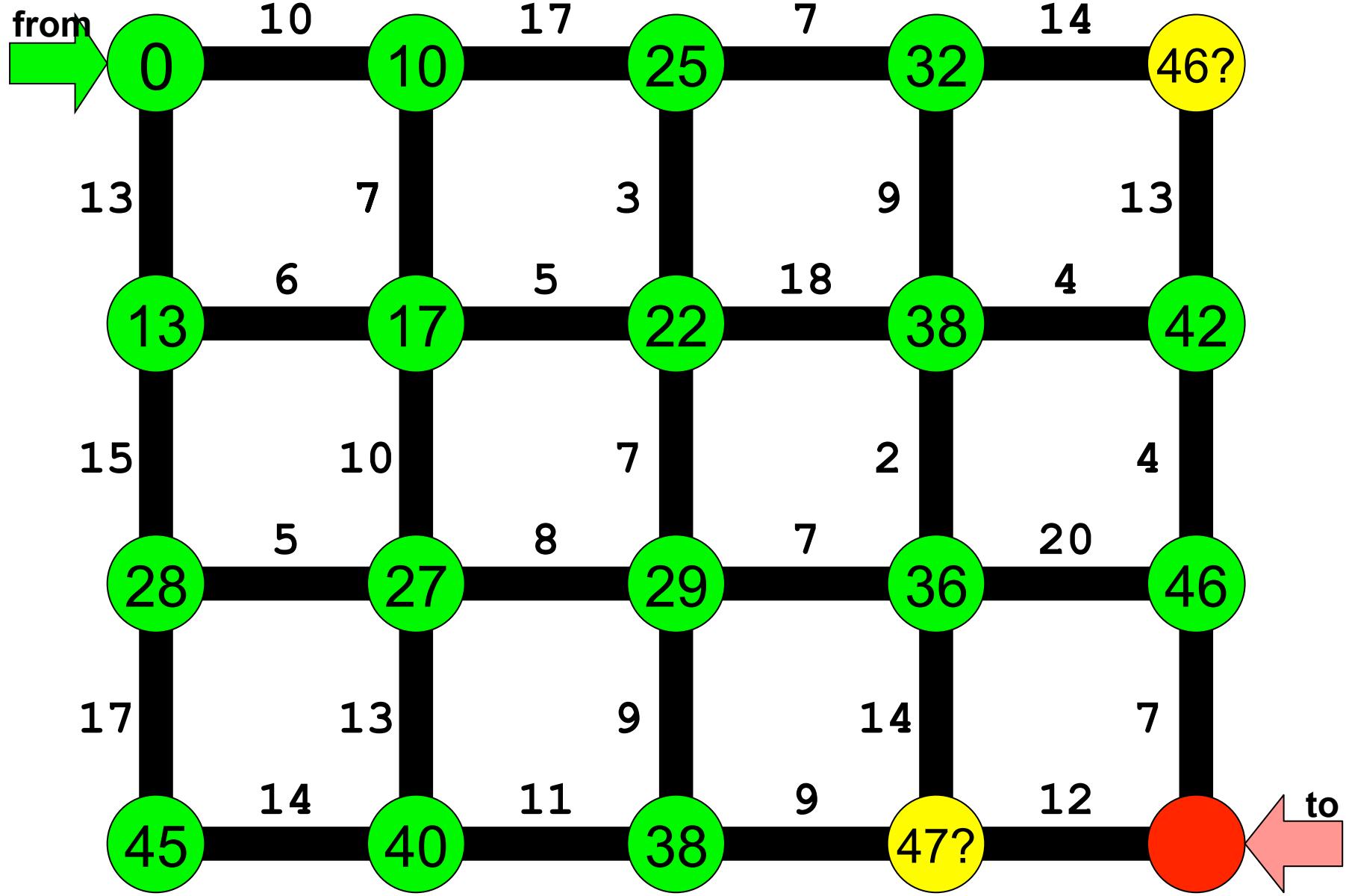


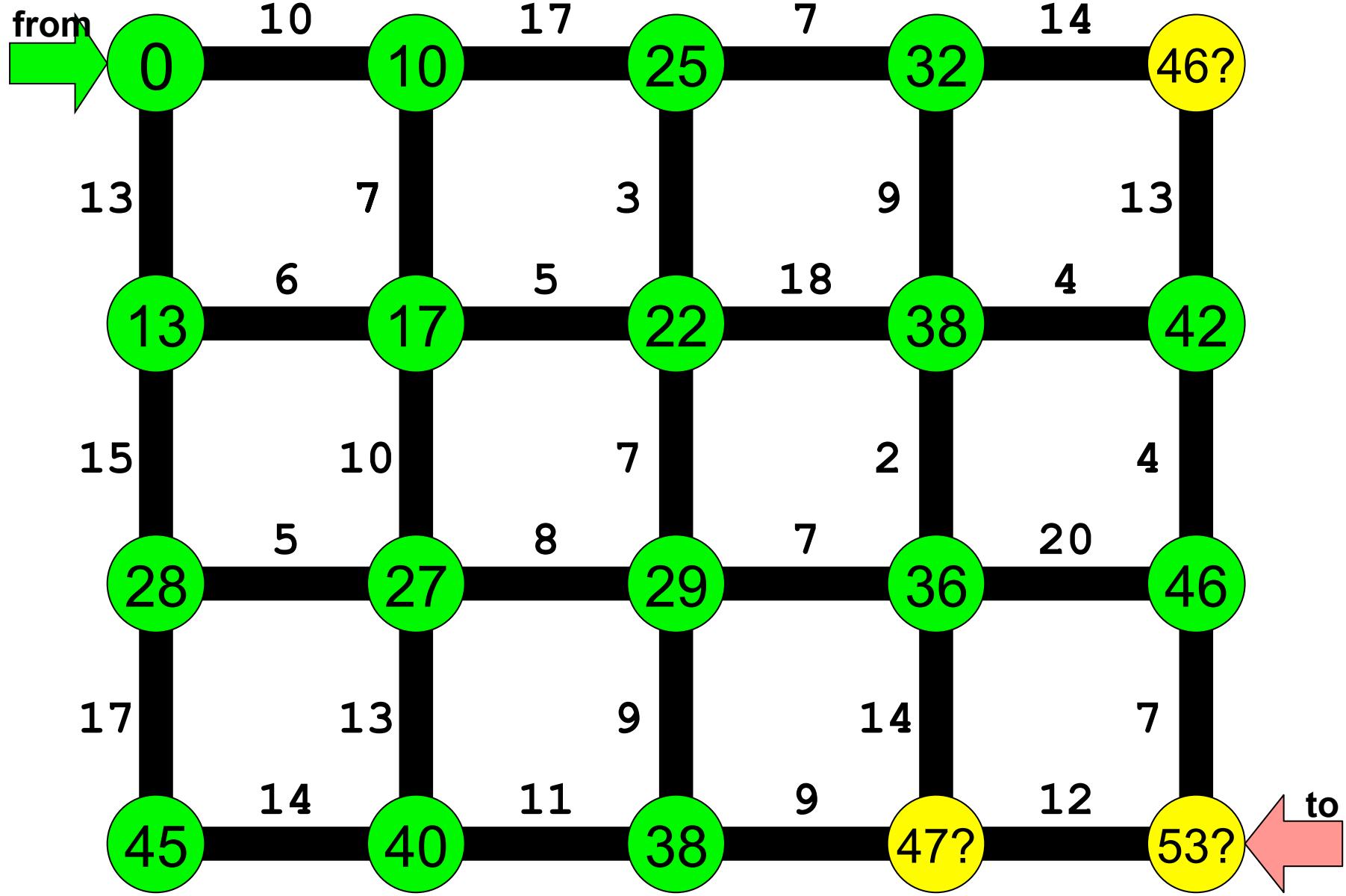


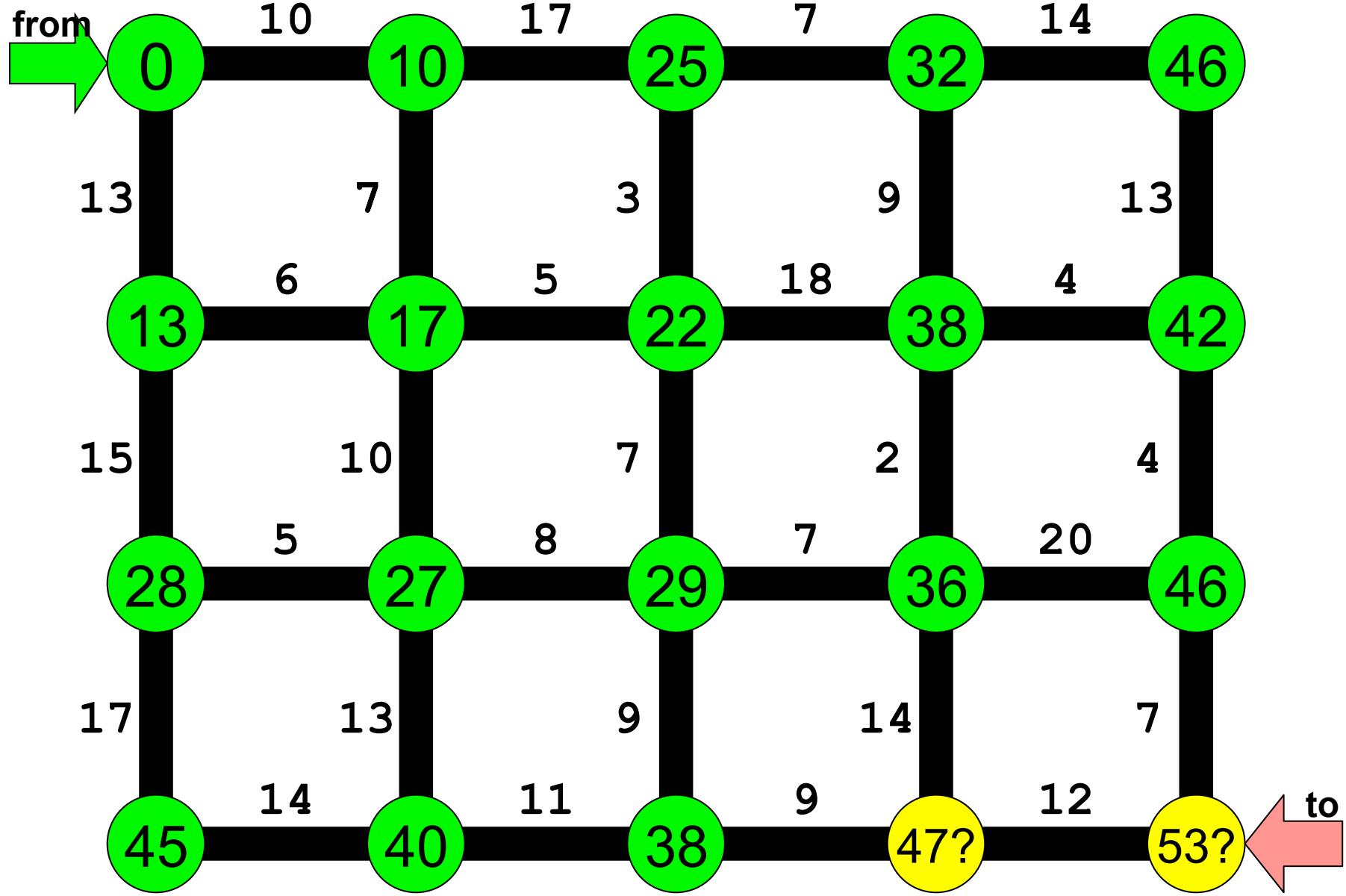


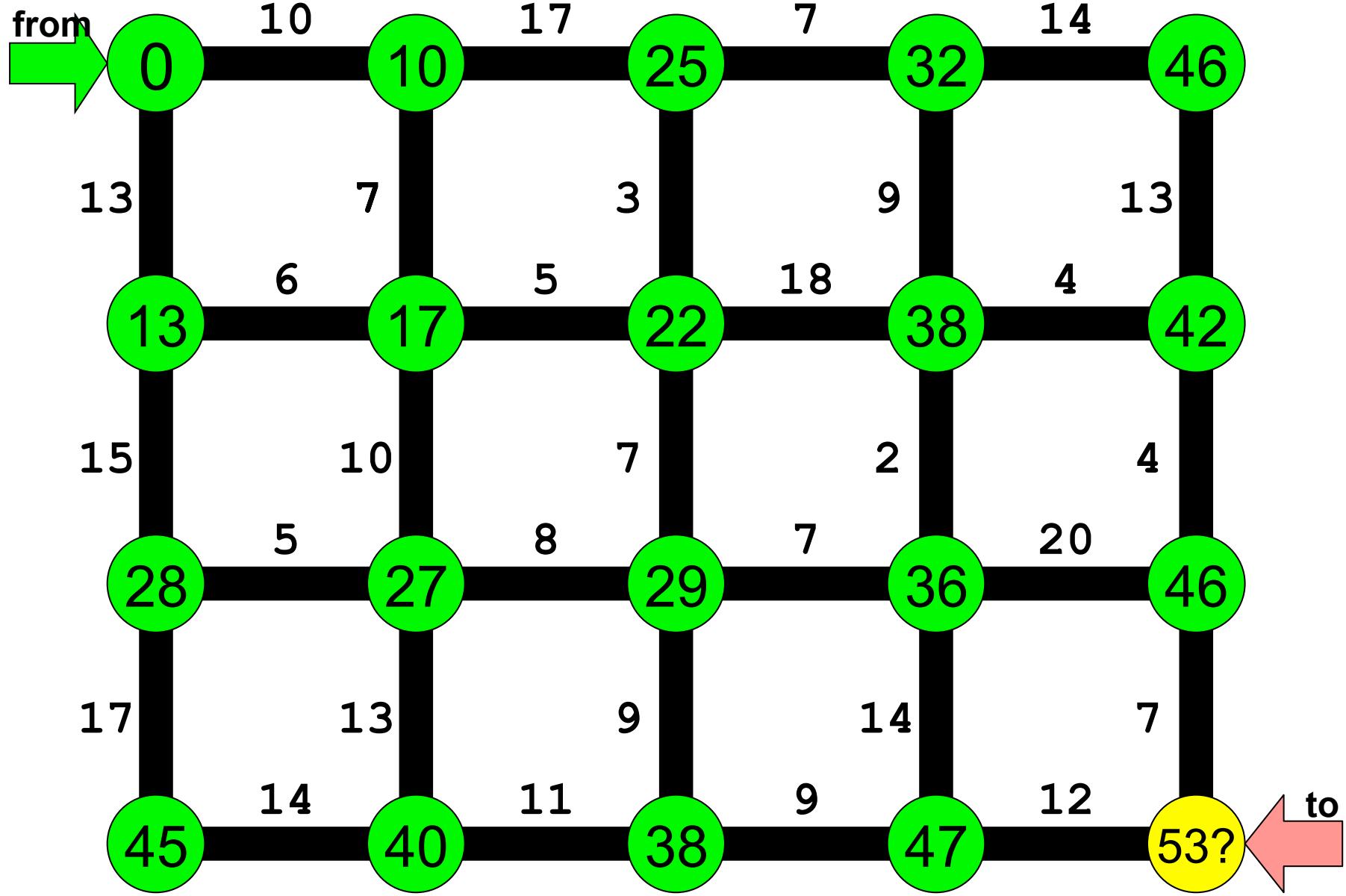


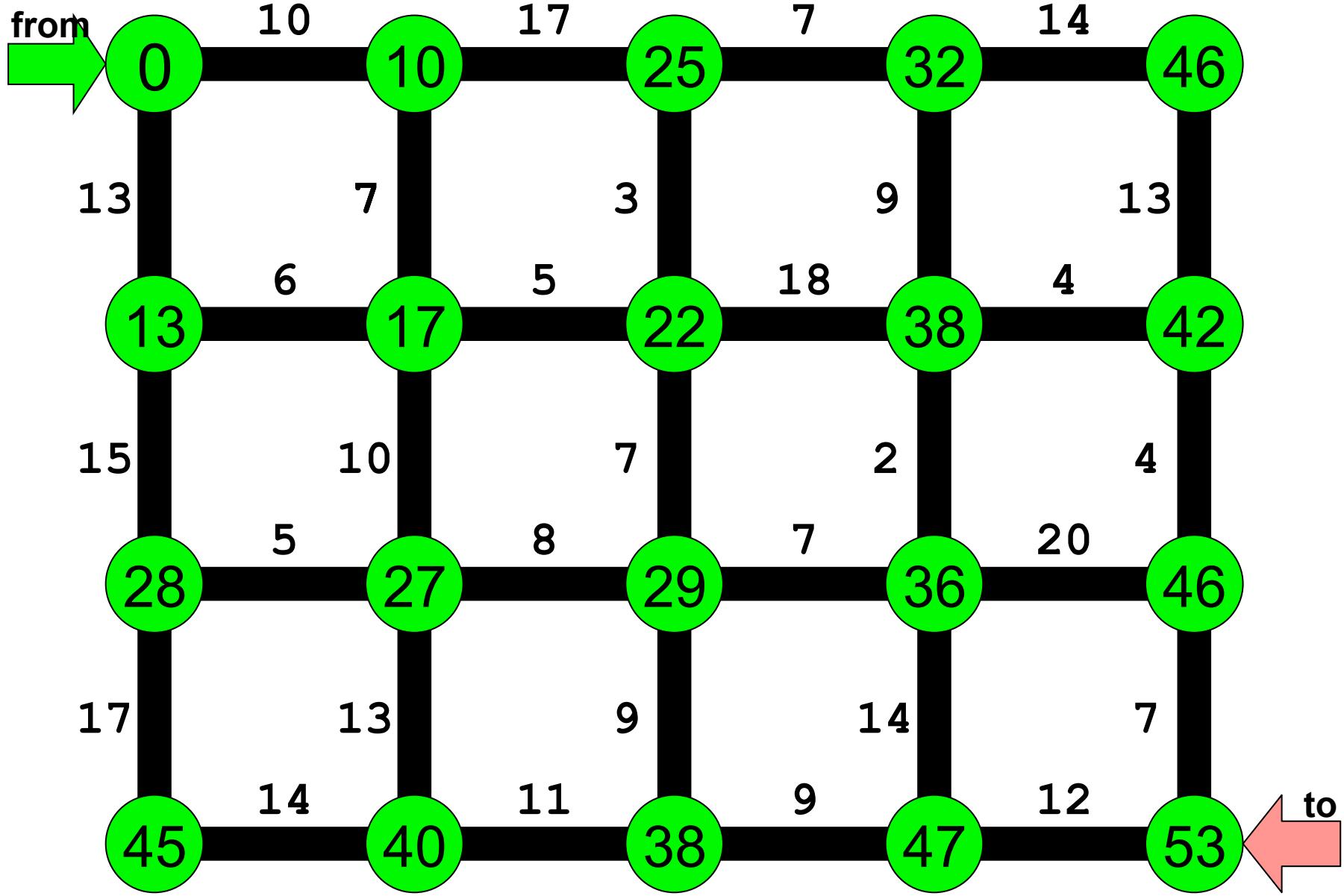


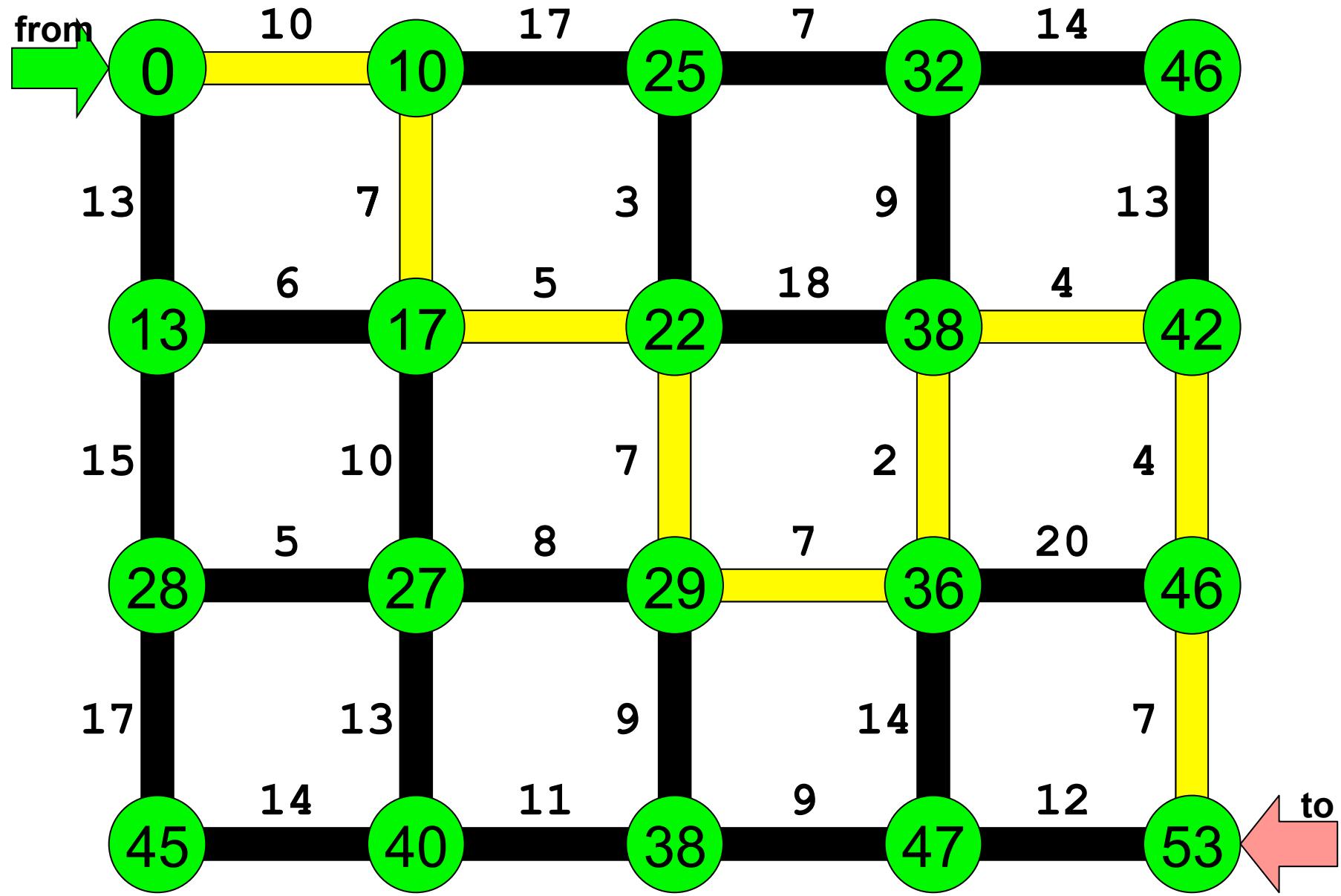


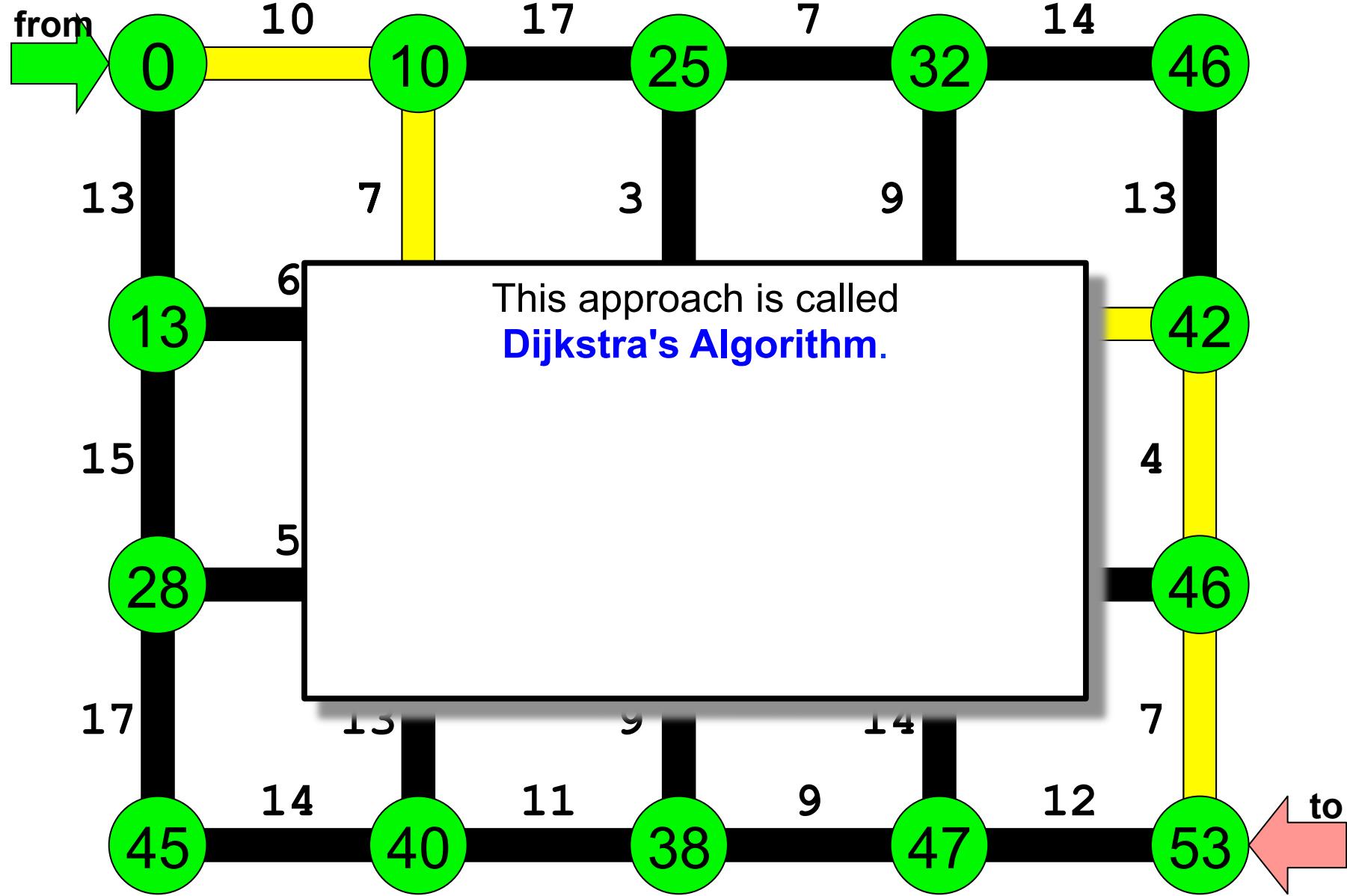


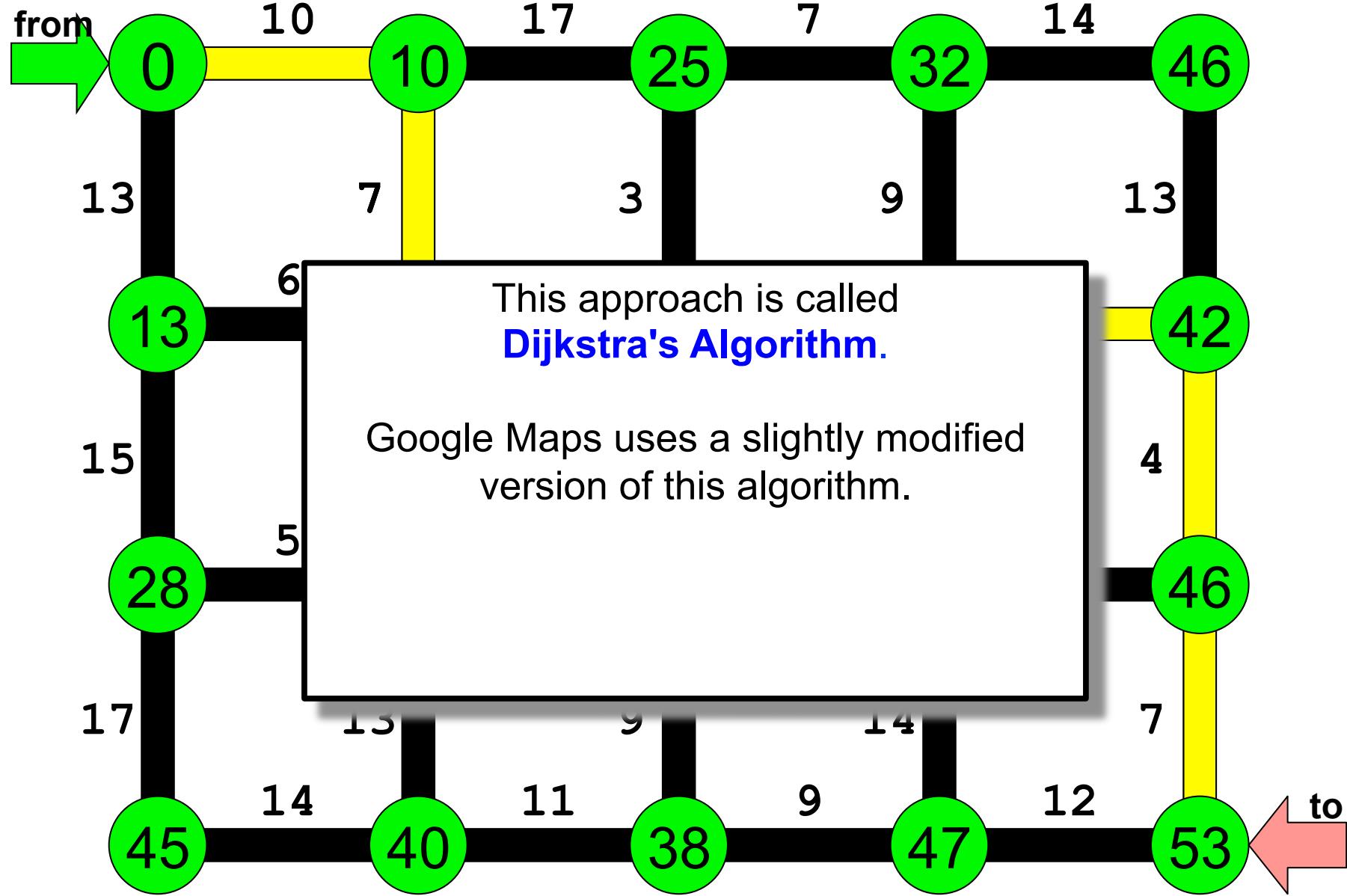


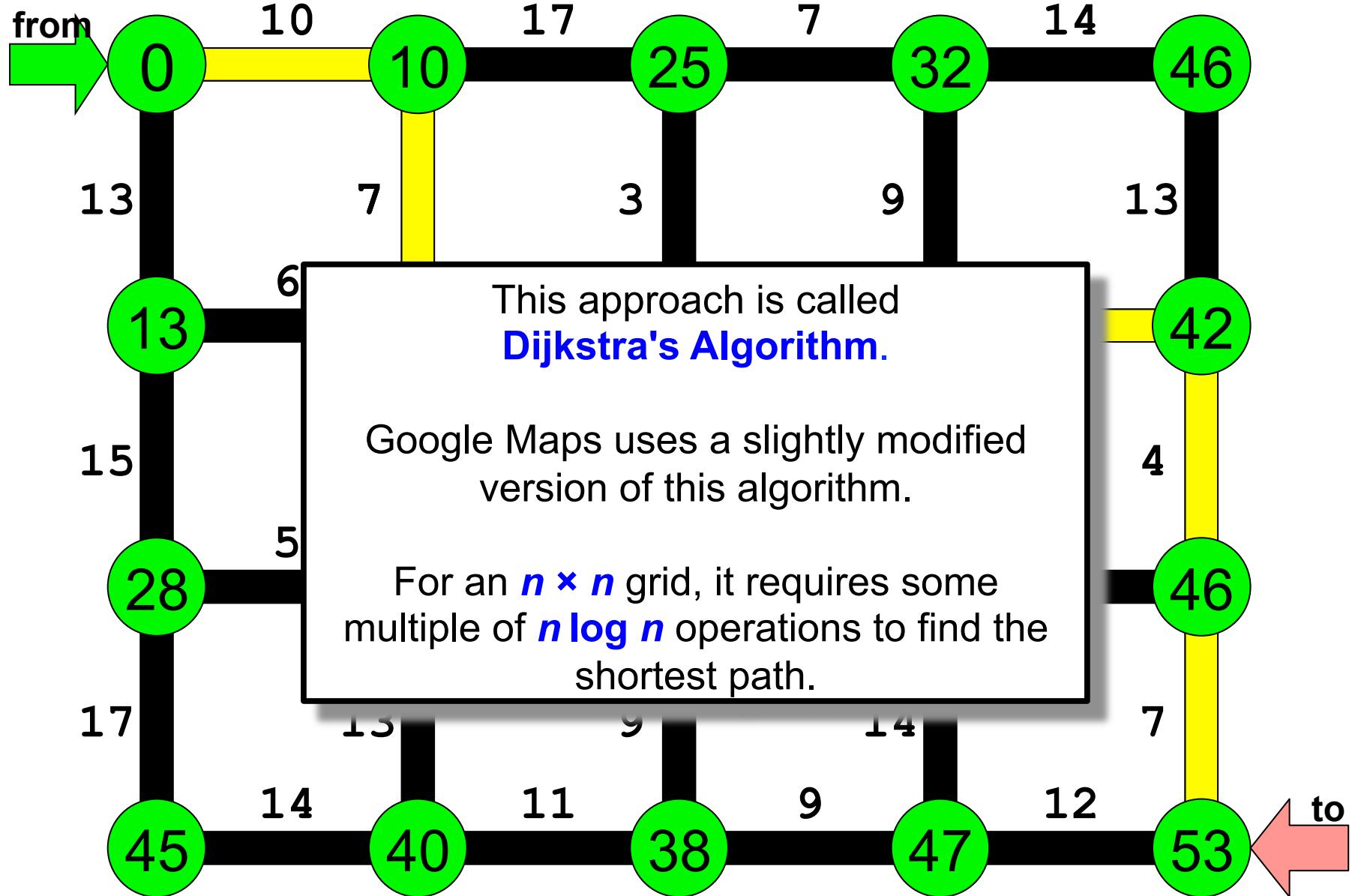










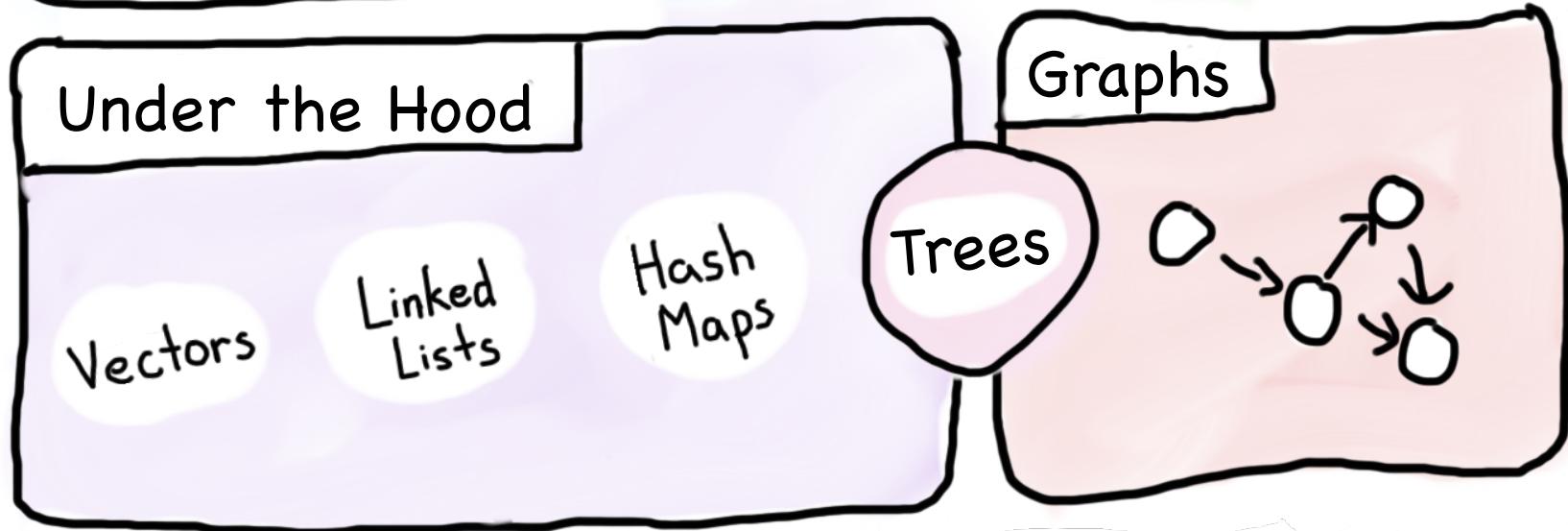
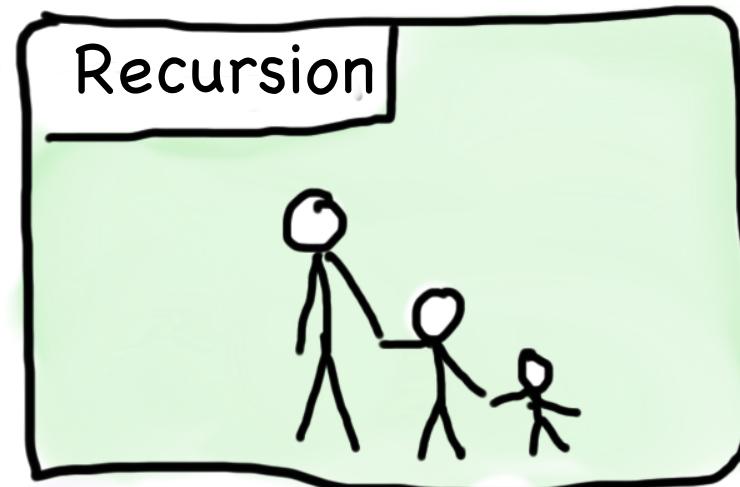
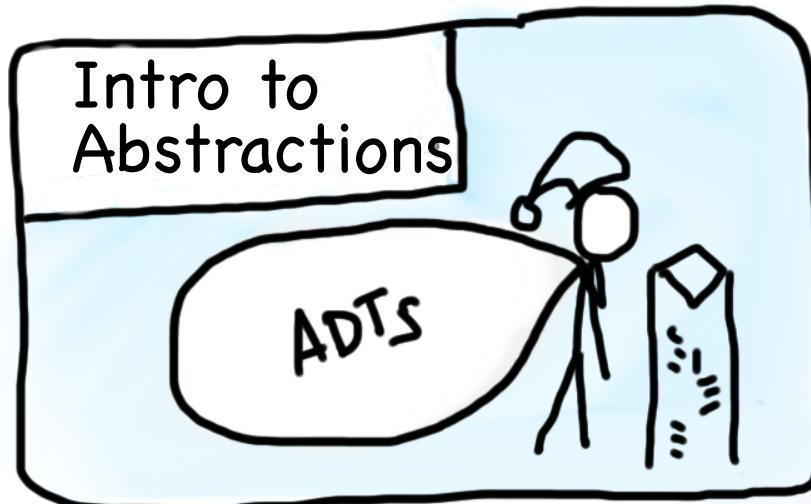


# Goals for the Course

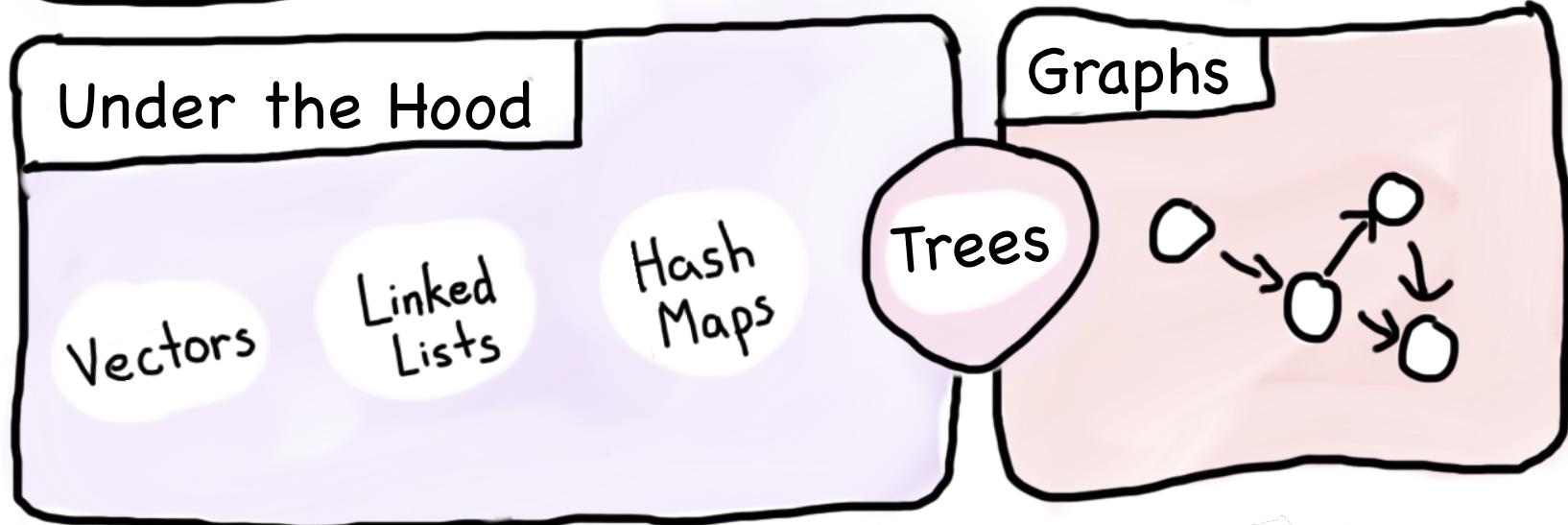
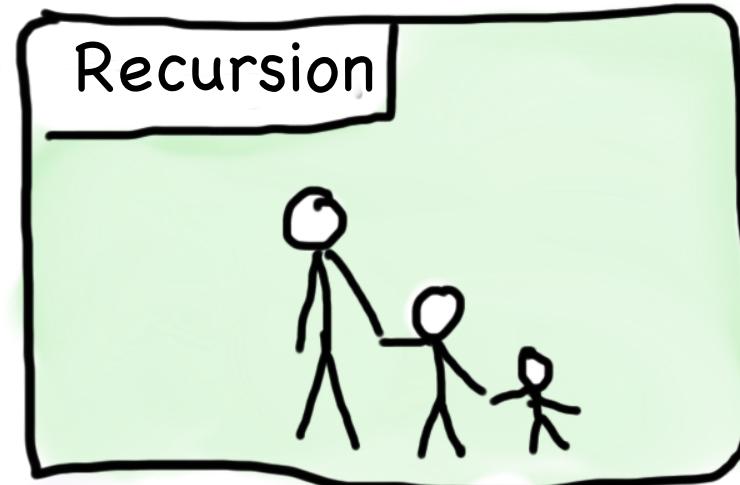
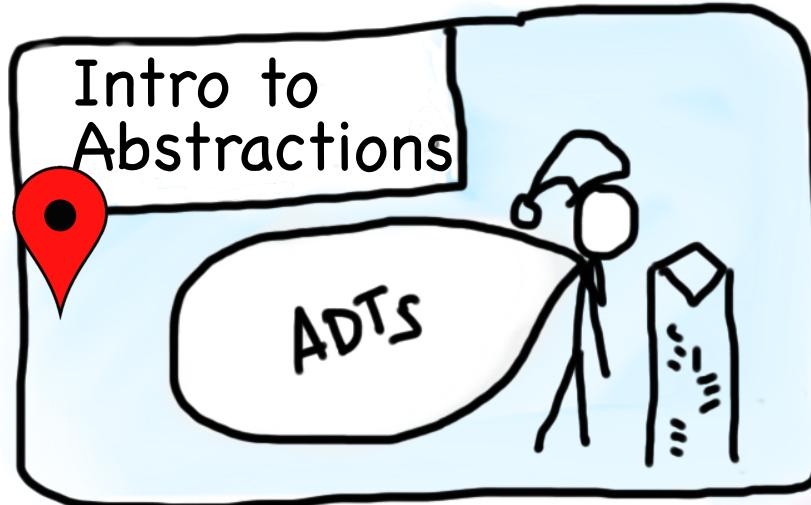
- **Learn how to model and solve complex problems with computers.**
- To that end:
  - Explore common abstractions for representing problems.
  - Harness recursion and understand how to think about problems recursively.
  - Learn and analyze different approaches for solving problems.

# Course Syllabus

# Course Syllabus



# Course Syllabus



You are here

# Is CS106B the Right Class?

## CS 106A: Programming Methodology (ENGR 70A)

Introduction to the engineering of computer applications emphasizing modern software engineering principles: object-oriented design, decomposition, encapsulation, abstraction, and testing. Uses the Java programming language. Emphasis is on good programming style and the built-in facilities of the Java language. No prior programming experience required.

**Terms:** Aut, Win, Spr, Sum | **Units:** 3-5 | **UG Reqs:** GER:DBEngrAppSci | **Grading:** Letter or Credit/No Credit

**Instructors:** Burr, B. (PI); Cain, G. (PI); Cooper, S. (PI); Jimenez, O. (PI); Sahami, M. (PI)

## CS 106X: Programming Abstractions (Accelerated) (ENGR 70X)

Intensive version of 106B for students with a strong programming background interested in a rigorous treatment of the topics at an accelerated pace. Additional advanced material and more challenging projects.

Prerequisite: excellence in 106A or equivalent, or consent of instructor.

**Terms:** Aut, Win | **Units:** 3-5 | **UG Reqs:** GER:DBEngrAppSci | **Grading:** Letter or Credit/No Credit

**Instructors:** Cain, G. (PI)

One last detail

C++

Logistics

## Course Overview

First Program

Excited?

# Logistics

## Course Overview

### First Program

### Excited?

# Ready to Dive In?



# Hello World

```
/*
 * The canonical hello world program in C++
 */
```

```
#include <iostream> // A true classic.
#include "console.h" // A Stanford Library!
using namespace std; // The best namespace.

int main() {
    cout << "hello, world!" << endl;
    return 0;
}
```

# Hello World

```
/*
 * The canonical hello world program in C++
 */
```

```
#include <iostream> // A true classic.
#include "console.h" // A Stanford Library!
using namespace std; // The best namespace.
```

```
int main() {
    cout << "hello, world!" << endl;
    return 0;
}
```

# Hello World

```
/*
 * The canonical hello world program in C++
 */

#include <iostream> // A true classic.
#include "console.h" // A Stanford Library!
using namespace std; // The best namespace.

int main() {
    cout << "hello, world!" << endl;
    return 0;
}
```

# Hello World

```
/*
 * The canonical hello world program in C++
 */

#include <iostream> // A true classic.
#include "console.h" // A Stanford Library!
using namespace std; // The best namespace.

int main() {
    cout << "hello, world!" << endl;
    return 0;
}
```

# Logistics

## Course Overview

### First Program

### Excited?

# Logistics

## Course Overview

### First Program

Excited?

# Why CS?



photo: D Sallery

# Why CS?



# Why CS?



# Why CS?



The End.

A close-up photograph of a shiny, metallic silver top hat. The hat is positioned diagonally, with its crown pointing towards the top left and its base towards the bottom right. It sits on a dark, textured surface that appears to be a book cover or endpaper, showing some wear and discoloration. The lighting is dramatic, coming from the side to create strong highlights on the curved edges and deep shadows in the interior of the hat.

The End?