Welcome to CS106B!

- Five Handouts
  - Course information
  - Syllabus
  - Course placement information
  - Honor Code policies
  - Assignment 0: Welcome to CS106B!
- Today:
  - Course Overview
  - Where are We Going?
  - Introduction to C++

Who's Here Today?

- Aeronautics and Astronautics
- Biochemistry
- Bioengineering
- Biology
- Biomedical Informatics
- Business Administration
- Chemical Engineering
- Chemistry
- Chinese
- Civil and Environmental Engineering
- Computational and Mathematical Engineering
- Computer Science
- Creative Writing
- East Asian Studies
- Economics
- Electrical Engineering
- Energy Resources Engineering
- Engineering
- Environment and Resources
- Feminism, Gender, and Sexuality Studies
- Film and Media Studies
- German Studies
- Human Biology
- Immunology
- International Policy Studies
- Law
- Management Science and Engineering
- Materials Science and Engineering
- Mathematical and Computational Sciences
- Mechanical Engineering
- Medicine
- Music
- Petroleum Engineering
- Physics
- Political Science
- Psychology
- Public Policy
- Science, Technology, and Society
- Statistics
- Stem Cell Biology and Regenerative Medicine
- Symbolic Systems
- Theater and Performing Studies
- ***Undeclared!***

Course Staff

Instructor: Keith Schwarz
(htiek@cs.stanford.edu)

Head TA: Anton Apostolatos
(antonaf@stanford.edu)

The CS106B Section Leaders
The CS106B Course Helpers

Course Website

http://cs106b.stanford.edu

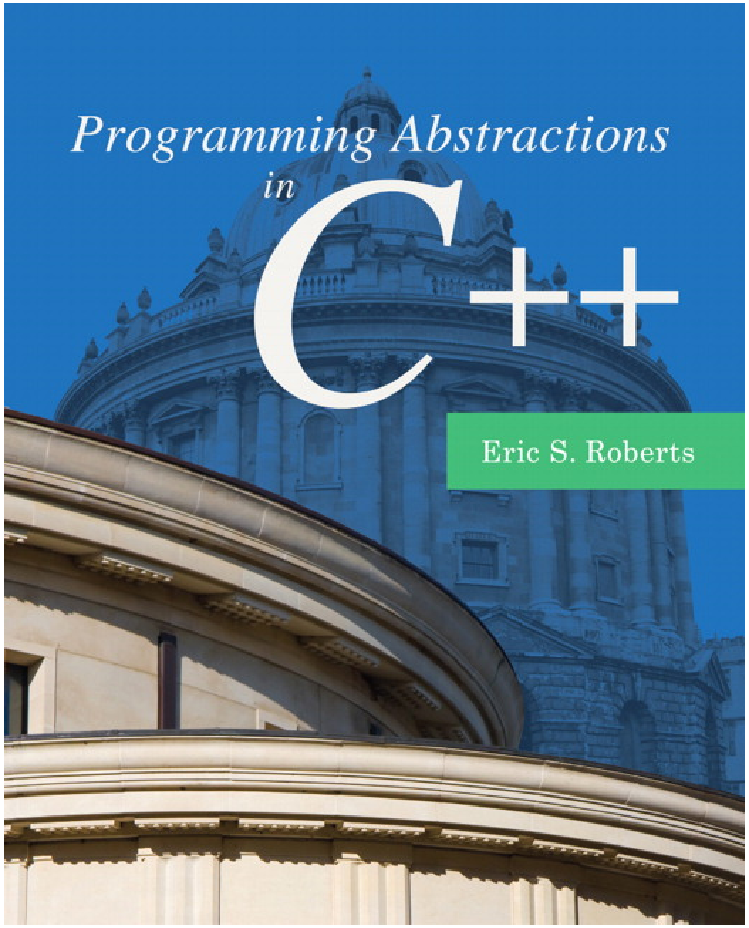
Prerequisites

CS106A

(or equivalent)

(check out our course placement handout if you're unsure!)

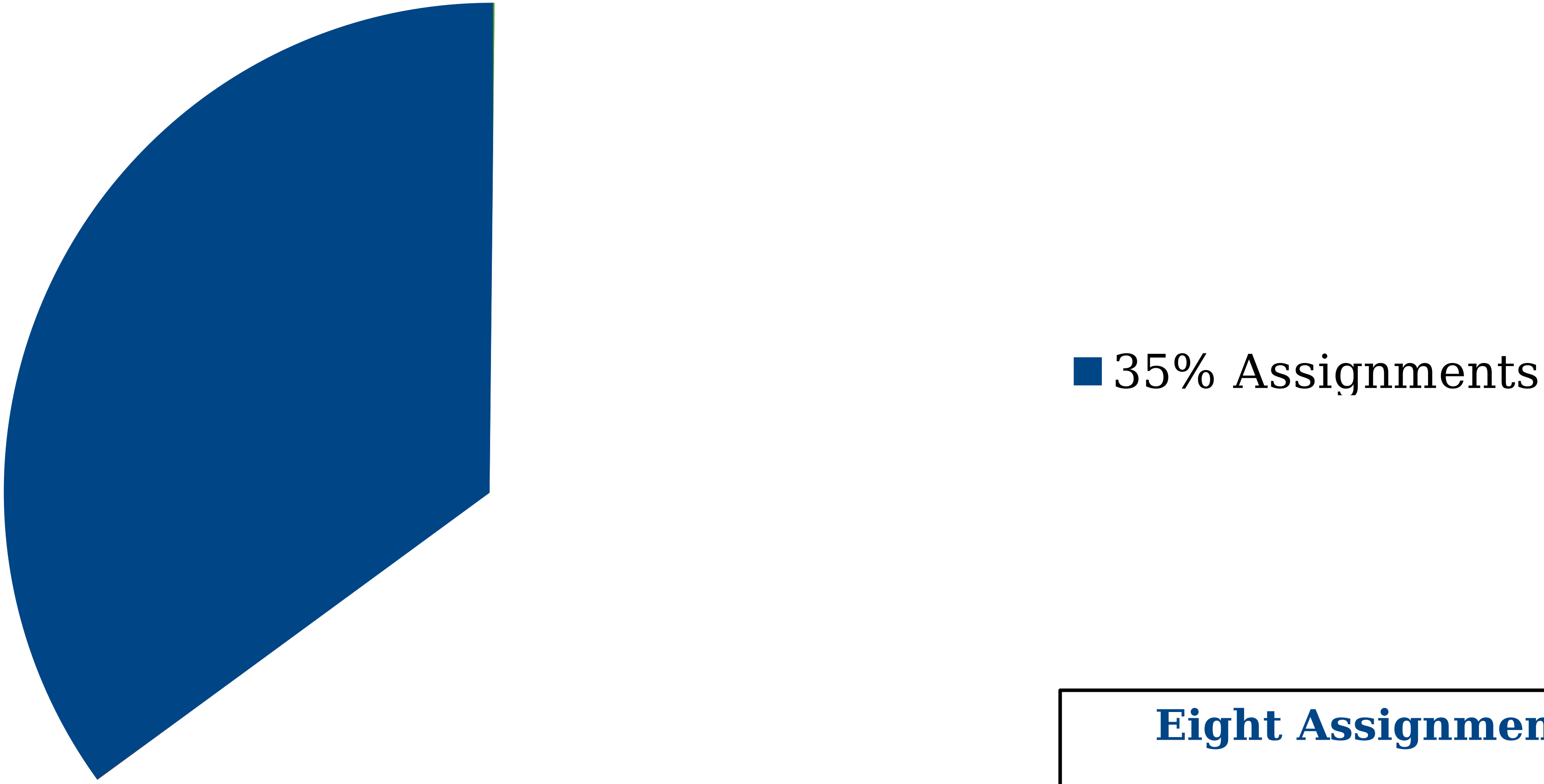
Required Reading



- Available in the bookstore. Some copies are on reserve in the Engineering library.
- There are (old, outdated) PDFs floating around; use them at your own risk!
- We do recommend picking up a copy of this book, since it provides a lot of useful extra background information.

Grading Policies

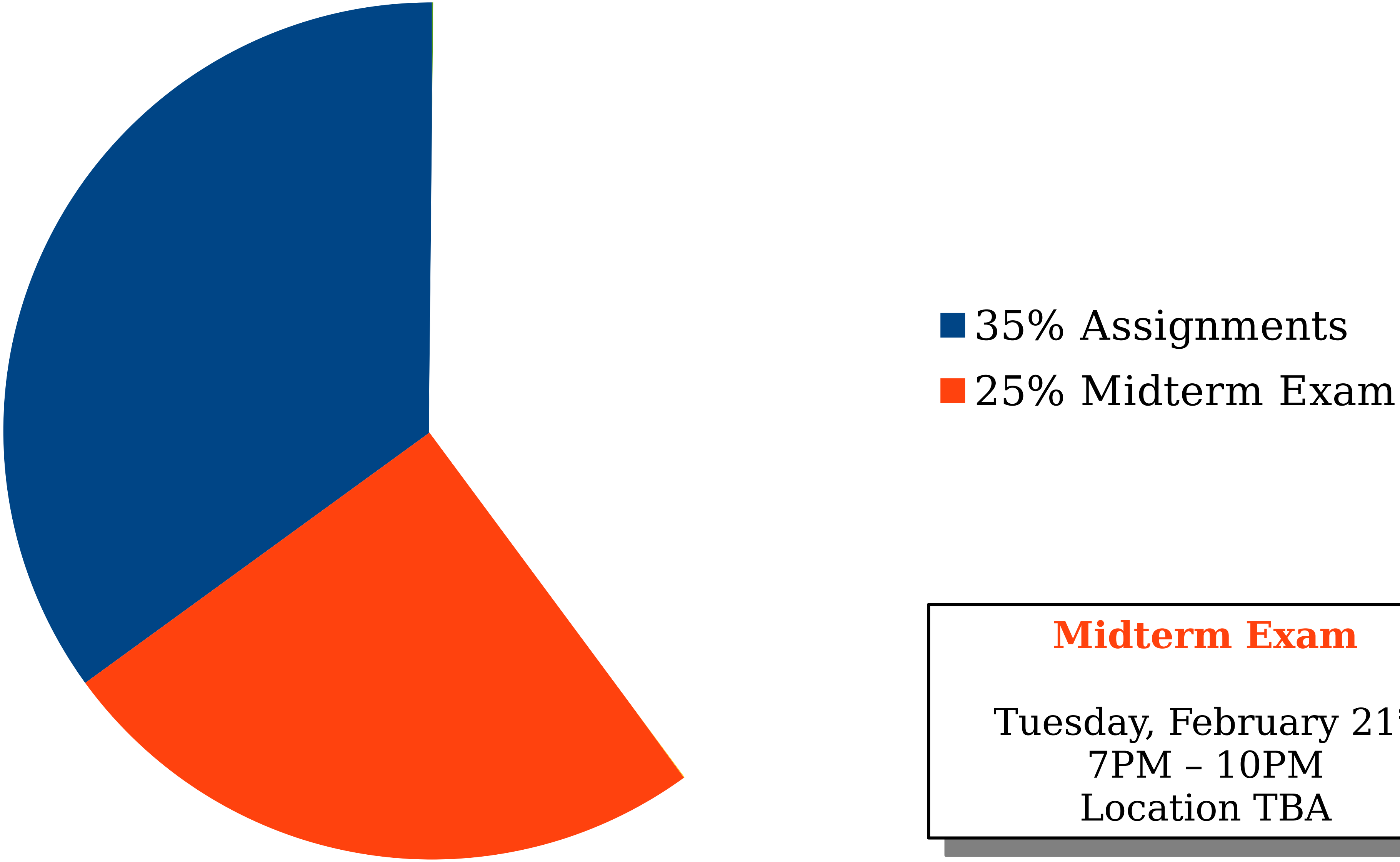
Grading Policies



Eight Assignments

(One intro assignment that goes out today, seven programming assignments)

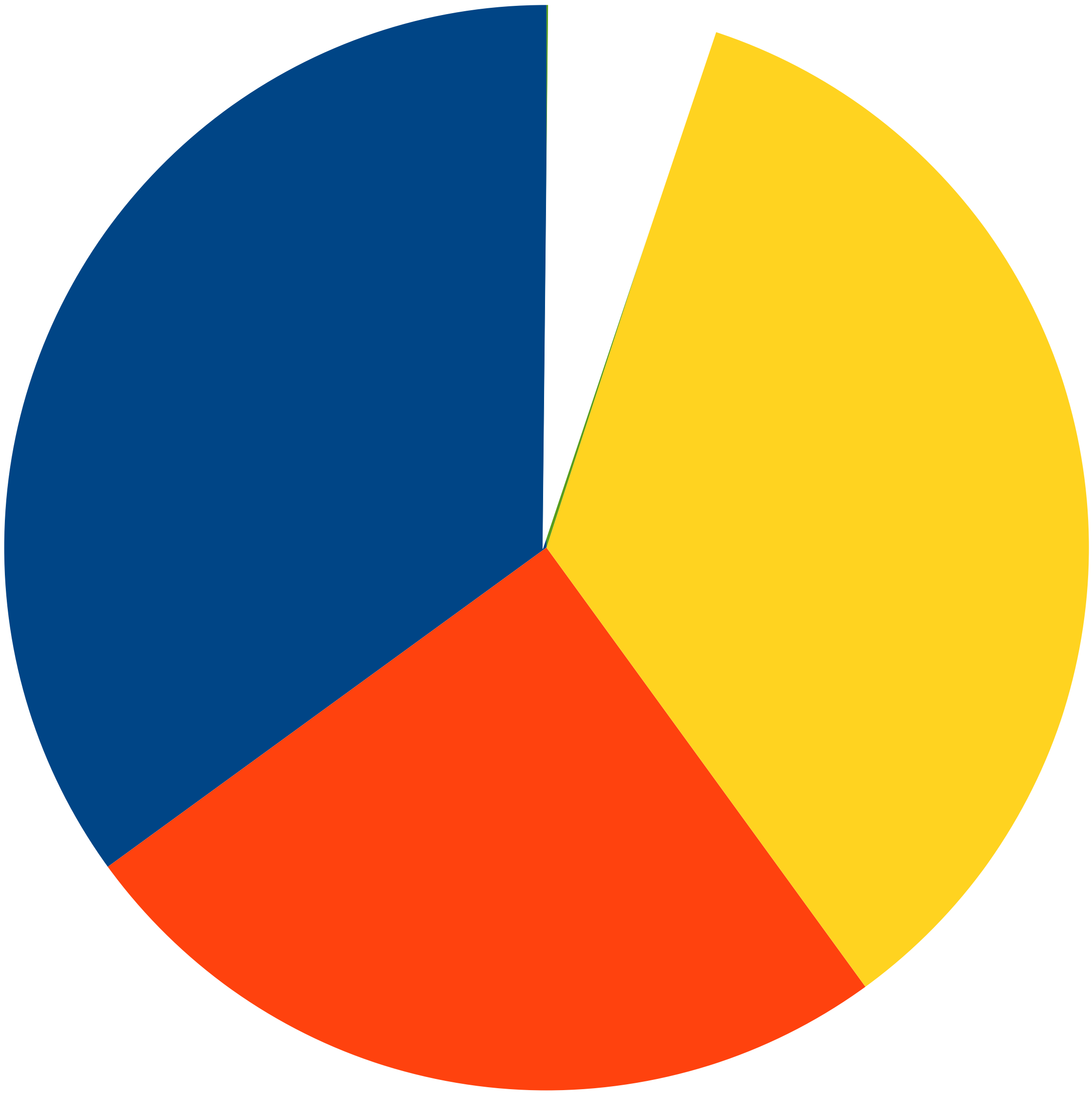
Grading Policies



Midterm Exam

Tuesday, February 21st
7PM – 10PM
Location TBA

Grading Policies

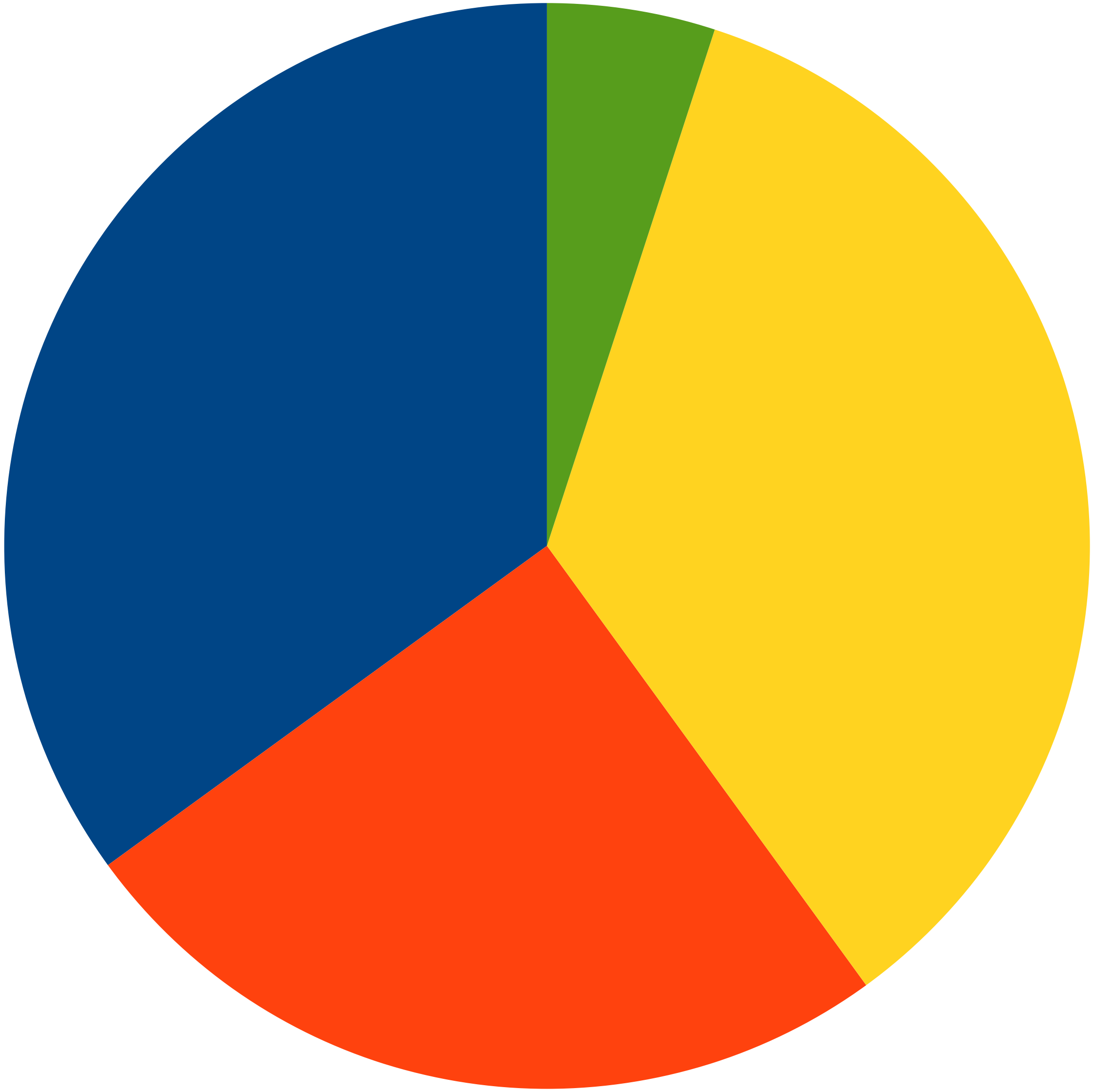


Final Exam

Monday, March 20th
8:30AM – 11:30AM

No alternate exams except for OAE accommodations.

Grading Policies



Discussion Sections

Weekly sections. Let’s go talk about them!

Discussion Sections

- There are weekly discussion sections in CS106B. Section attendance is required.
- Sign up between Thursday, January 12th at 5:00PM and Sunday, January 16th at 5:00PM by visiting

http://cs198.stanford.edu/section

- We don’t look at Axess for section enrollments. Please make sure to sign up here even if you’re already enrolled on Axess.

How Many Units?

```
int numUnits(bool isGrad) {
    if (isGrad) {
        return randomInteger(3, 5); // 3 to 5
    } else {
        return 5;
    }
}
```

Getting Help

Getting Help

- LaIR Hours!
  - Sunday – Thursday, 6PM – Midnight
  - Starts next week.
- Anton's Office Hours in the Huang basement
  - Wednesdays, 2:00PM – 4:00PM
- Keith's Office Hours in Gates 219
  - Tuesdays, 2:15PM – 4:15PM (starting next week)
  - Come hang out and chat about whatever it is that you’re interested in!

What's Next in Computer Science?

Goals for this 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.
  - Quantitatively analyze different approaches for solving problems.

Goals for this Course

Learn how to model and solve complex problems with computers.

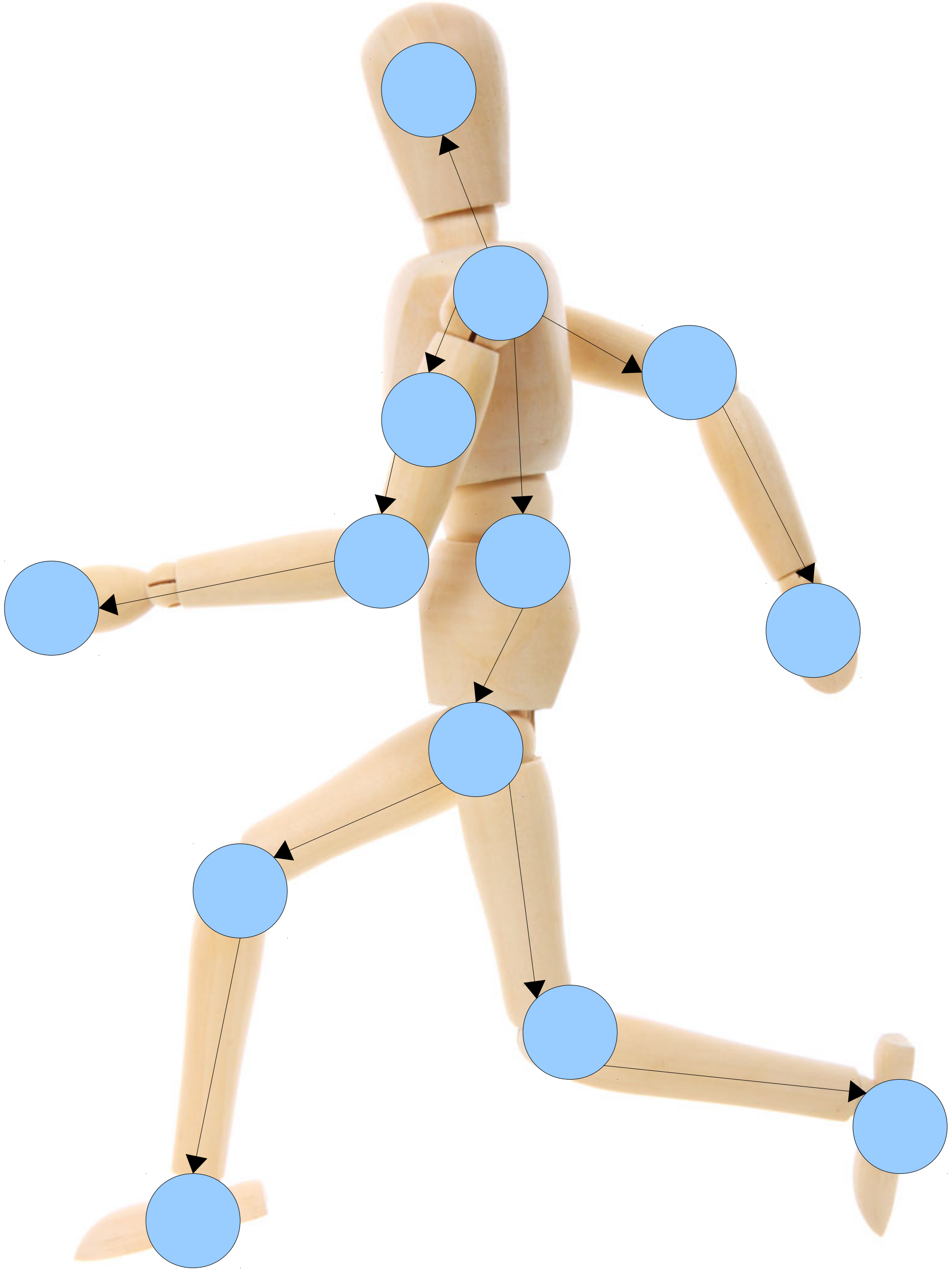
To that end:

- Explore common abstractions for representing problems.

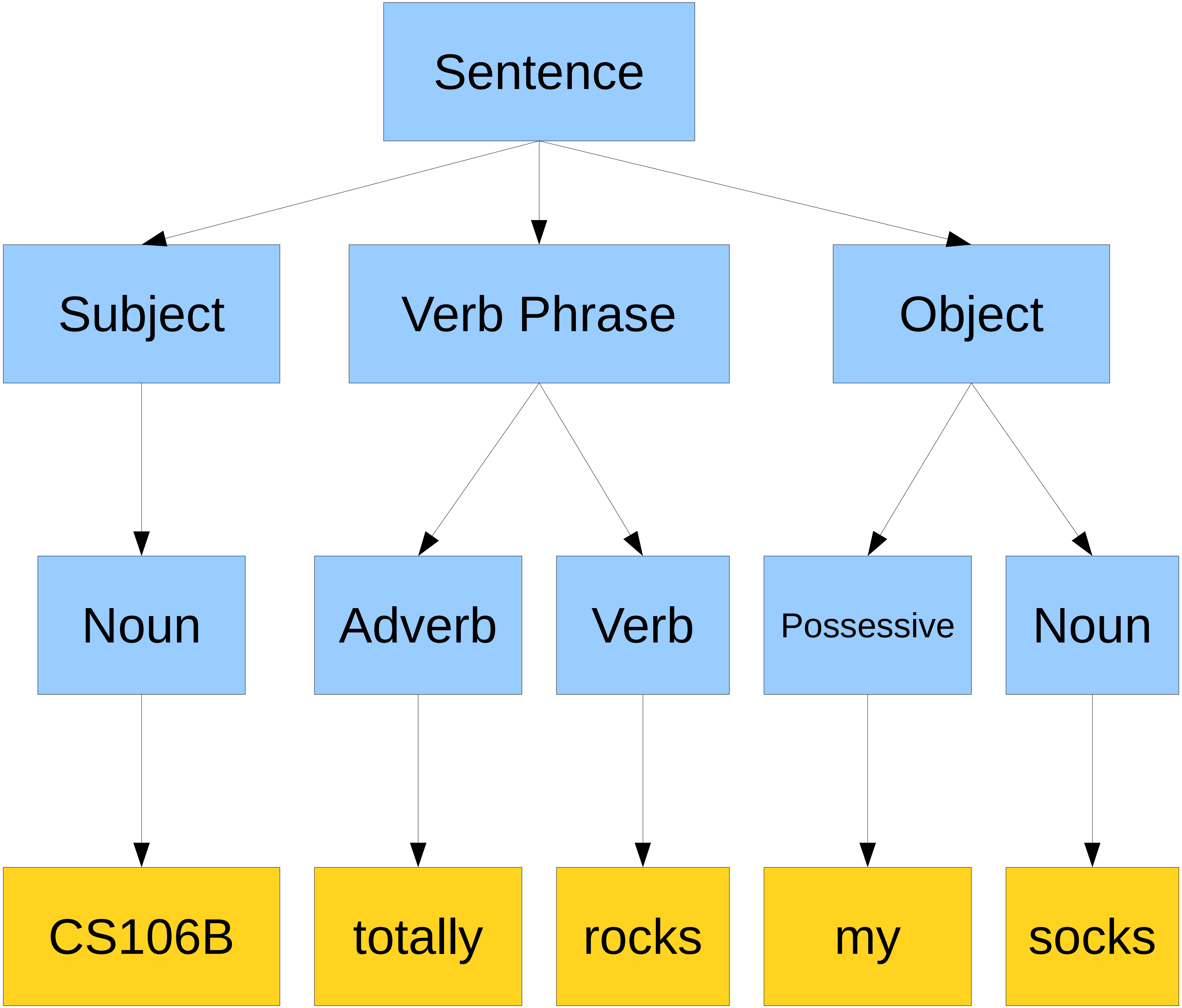
Harness recursion and understand how to think about problems recursively.

Quantitatively analyze different approaches for solving problems.

http://www.publicdomainpictures.net/pictures/10000/velka/1-1265899974oKJ9.jpg



http://www.publicdomainpictures.net/pictures/10000/velka/1-1265899974oKJ9.jpg



Sentence

Subject

Verb Phrase

Object

Noun

Adverb

Verb

Possessive

Noun

CS106B

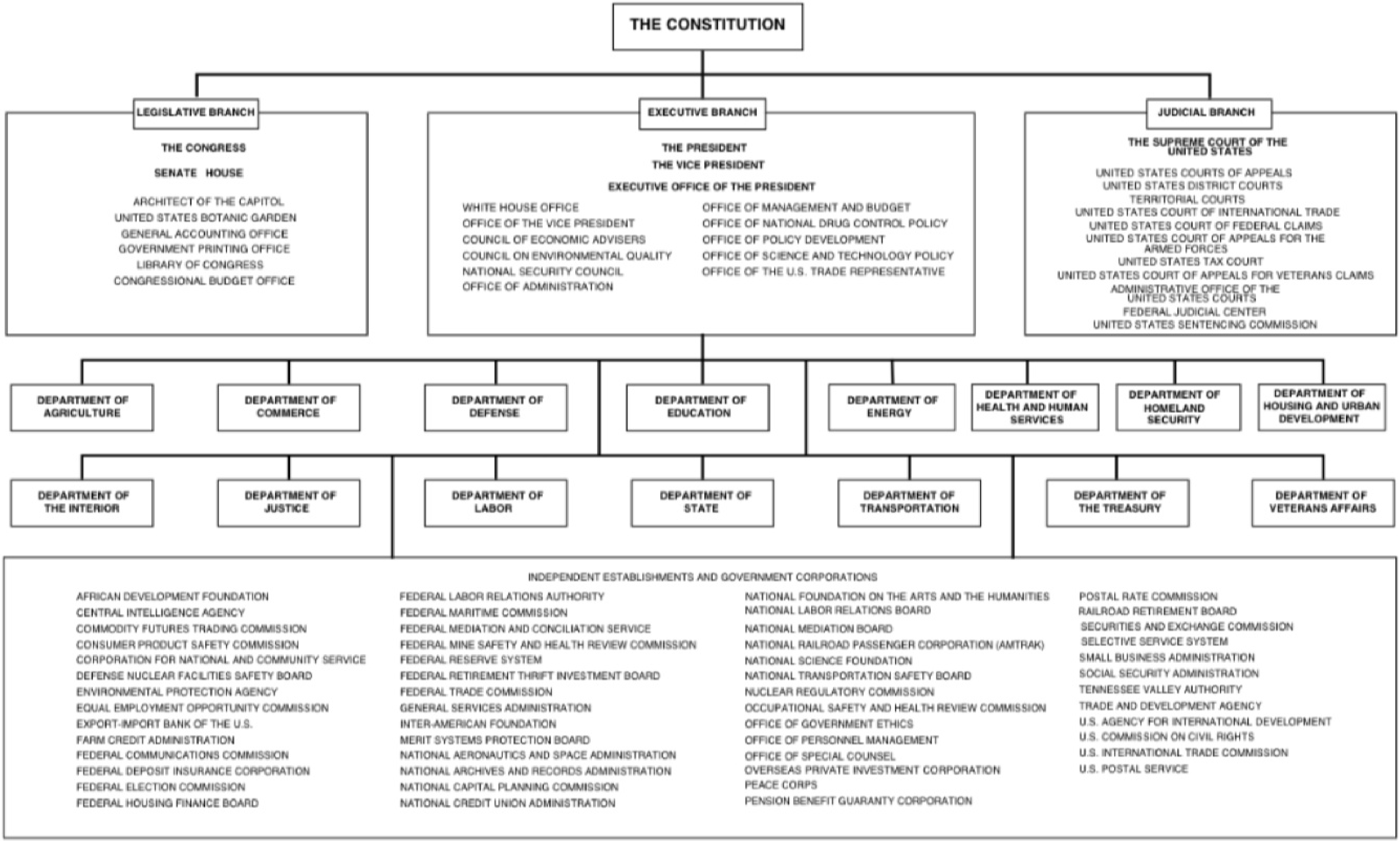
totally

rocks

my

socks

THE GOVERNMENT OF THE UNITED STATES



THE CONSTITUTION

LEGISLATIVE BRANCH

THE CONGRESS

SENATE HOUSE

ARCHITECT OF THE CAPITOL
UNITED STATES BOTANIC GARDEN
GENERAL ACCOUNTING OFFICE
GOVERNMENT PRINTING OFFICE
LIBRARY OF CONGRESS
CONGRESSIONAL BUDGET OFFICE

EXECUTIVE BRANCH

THE PRESIDENT
THE VICE PRESIDENT

EXECUTIVE OFFICE OF THE PRESIDENT

WHITE HOUSE OFFICE
OFFICE OF THE VICE PRESIDENT
COUNCIL OF ECONOMIC ADVISERS
COUNCIL ON ENVIRONMENTAL QUALITY
NATIONAL SECURITY COUNCIL
OFFICE OF ADMINISTRATION

OFFICE OF MANAGEMENT AND BUDGET
OFFICE OF NATIONAL DRUG CONTROL POLICY
OFFICE OF POLICY DEVELOPMENT
OFFICE OF SCIENCE AND TECHNOLOGY POLICY
OFFICE OF THE U.S. TRADE REPRESENTATIVE

JUDICIAL BRANCH

THE SUPREME COURT OF THE UNITED STATES

UNITED STATES COURTS OF APPEALS
UNITED STATES DISTRICT COURTS
TERRITORIAL COURTS
UNITED STATES COURT OF INTERNATIONAL TRADE
UNITED STATES COURT OF FEDERAL CLAIMS
UNITED STATES COURT OF APPEALS FOR THE ARMED FORCES
UNITED STATES TAX COURT
UNITED STATES COURT OF APPEALS FOR VETERANS CLAIMS
ADMINISTRATIVE OFFICE OF THE UNITED STATES COURTS
FEDERAL JUDICIAL CENTER
UNITED STATES SENTENCING COMMISSION

DEPARTMENT OF AGRICULTURE
DEPARTMENT OF COMMERCE
DEPARTMENT OF DEFENSE
DEPARTMENT OF EDUCATION
DEPARTMENT OF ENERGY
DEPARTMENT OF HEALTH AND HUMAN SERVICES
DEPARTMENT OF HOMELAND SECURITY
DEPARTMENT OF HOUSING AND URBAN DEVELOPMENT

DEPARTMENT OF THE INTERIOR
DEPARTMENT OF JUSTICE
DEPARTMENT OF LABOR
DEPARTMENT OF STATE
DEPARTMENT OF TRANSPORTATION
DEPARTMENT OF THE TREASURY
DEPARTMENT OF VETERANS AFFAIRS

INDEPENDENT ESTABLISHMENTS AND GOVERNMENT CORPORATIONS

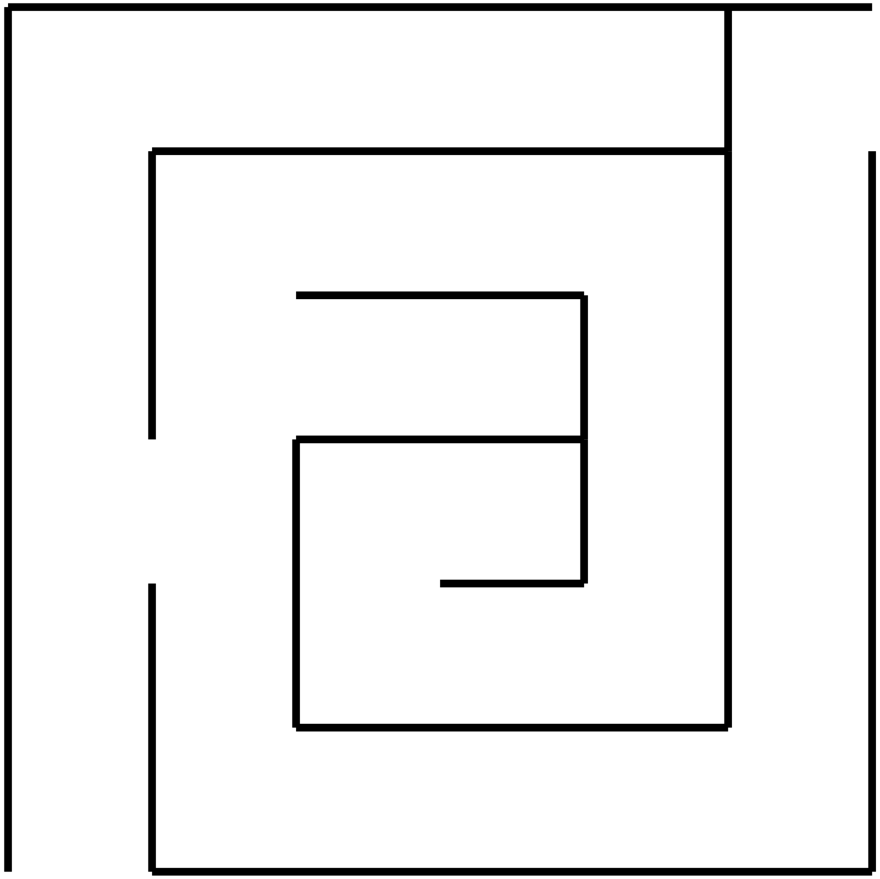
AFRICAN DEVELOPMENT FOUNDATION
CENTRAL INTELLIGENCE AGENCY
COMMODITY FUTURES TRADING COMMISSION
CONSUMER PRODUCT SAFETY COMMISSION
CORPORATION FOR NATIONAL AND COMMUNITY SERVICE
DEFENSE NUCLEAR FACILITIES SAFETY BOARD
ENVIRONMENTAL PROTECTION AGENCY
EQUAL EMPLOYMENT OPPORTUNITY COMMISSION
EXPORT-IMPORT BANK OF THE U.S.
FARM CREDIT ADMINISTRATION
FEDERAL COMMUNICATIONS COMMISSION
FEDERAL DEPOSIT INSURANCE CORPORATION
FEDERAL ELECTION COMMISSION
FEDERAL HOUSING FINANCE BOARD

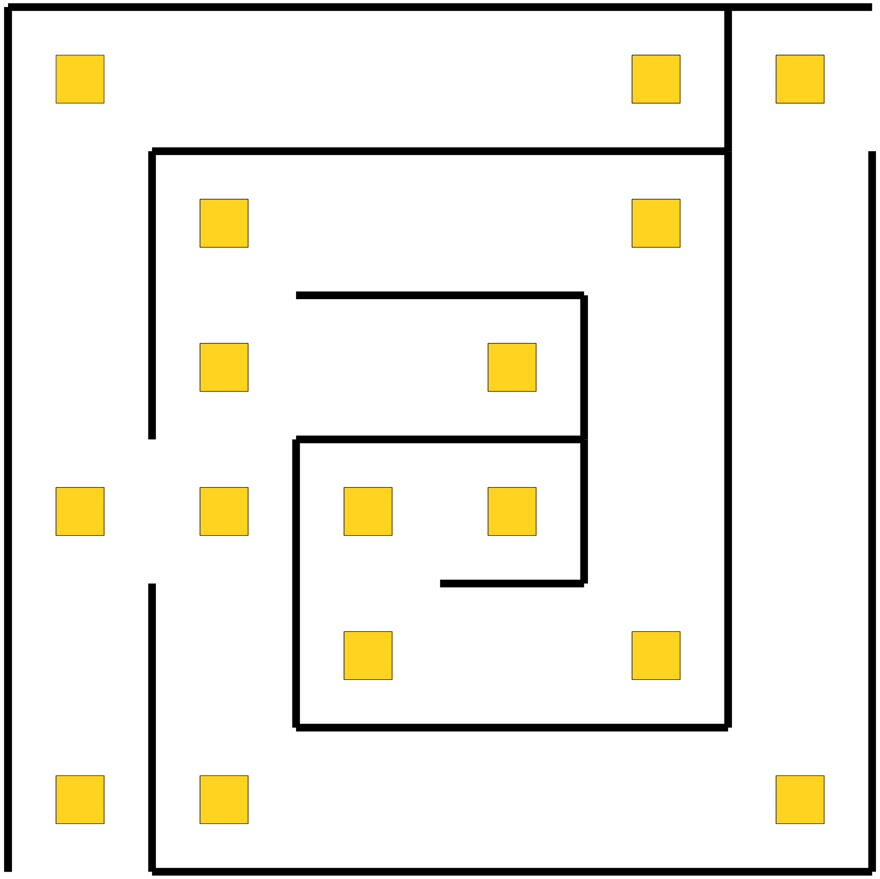
FEDERAL LABOR RELATIONS AUTHORITY
FEDERAL MARITIME COMMISSION
FEDERAL MEDIATION AND CONCILIATION SERVICE
FEDERAL MINE SAFETY AND HEALTH REVIEW COMMISSION
FEDERAL RESERVE SYSTEM
FEDERAL RETIREMENT THRIFT INVESTMENT BOARD
FEDERAL TRADE COMMISSION
GENERAL SERVICES ADMINISTRATION
INTER-AMERICAN FOUNDATION
MERIT SYSTEMS PROTECTION BOARD
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
NATIONAL ARCHIVES AND RECORDS ADMINISTRATION
NATIONAL CAPITAL PLANNING COMMISSION
NATIONAL CREDIT UNION ADMINISTRATION

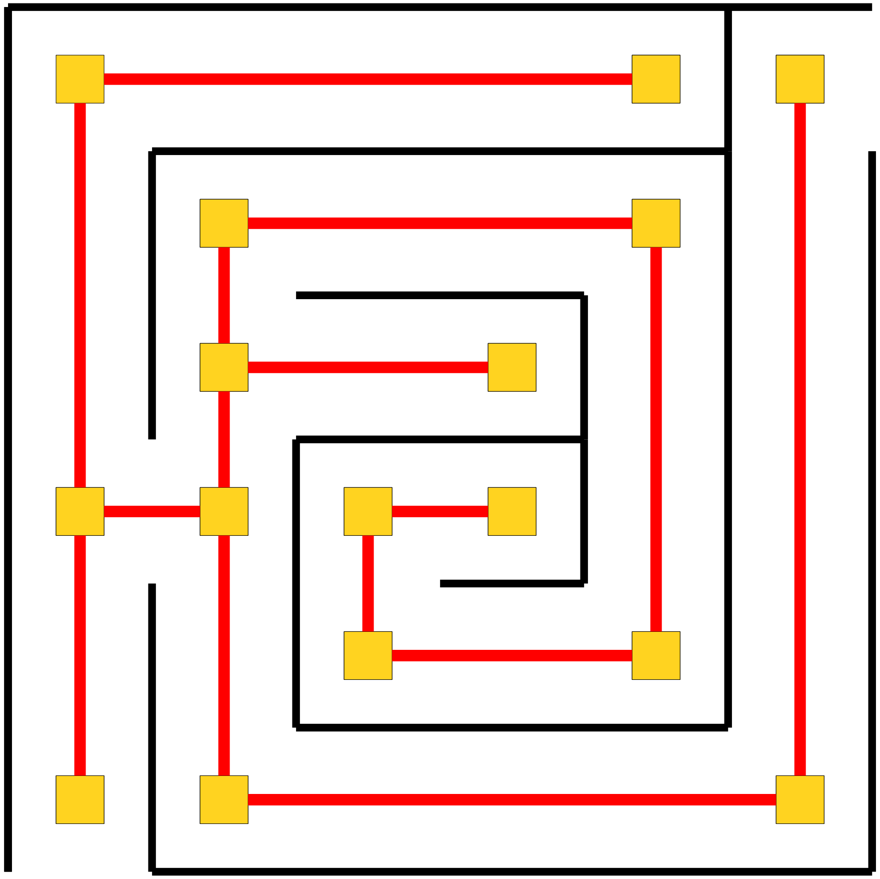
NATIONAL FOUNDATION ON THE ARTS AND THE HUMANITIES
NATIONAL LABOR RELATIONS BOARD
NATIONAL MEDIATION BOARD
NATIONAL RAILROAD PASSENGER CORPORATION (AMTRAK)
NATIONAL SCIENCE FOUNDATION
NATIONAL TRANSPORTATION SAFETY BOARD
NUCLEAR REGULATORY COMMISSION
OCCUPATIONAL SAFETY AND HEALTH REVIEW COMMISSION
OFFICE OF GOVERNMENT ETHICS
OFFICE OF PERSONNEL MANAGEMENT
OFFICE OF SPECIAL COUNSEL
OVERSEAS PRIVATE INVESTMENT CORPORATION
PEACE CORPS
PENSION BENEFIT GUARANTY CORPORATION

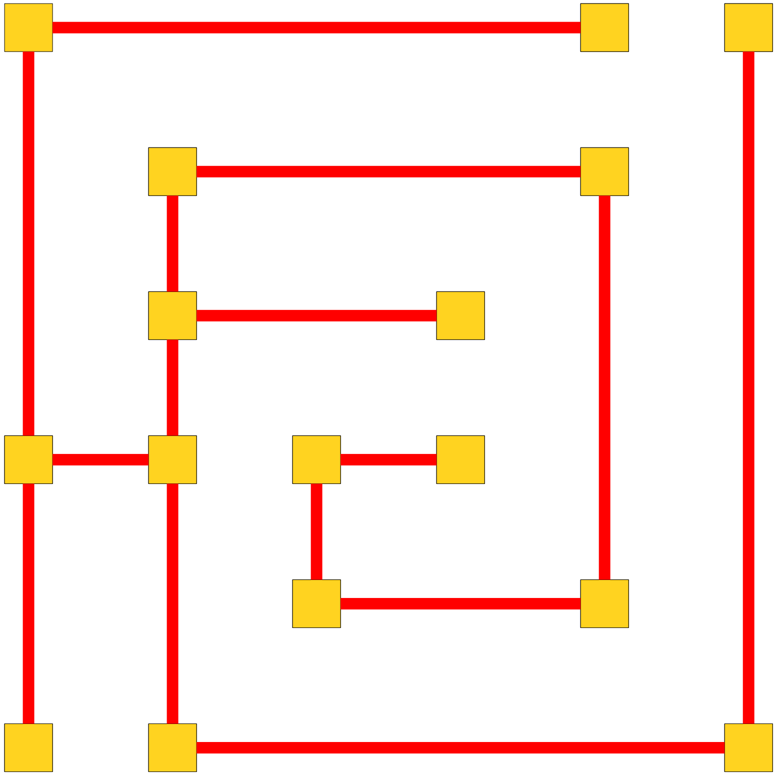
POSTAL RATE COMMISSION
RAILROAD RETIREMENT BOARD
SECURITIES AND EXCHANGE COMMISSION
SELECTIVE SERVICE SYSTEM
SMALL BUSINESS ADMINISTRATION
SOCIAL SECURITY ADMINISTRATION
TENNESSEE VALLEY AUTHORITY
TRADE AND DEVELOPMENT AGENCY
U.S. AGENCY FOR INTERNATIONAL DEVELOPMENT
U.S. COMMISSION ON CIVIL RIGHTS
U.S. INTERNATIONAL TRADE COMMISSION
U.S. POSTAL SERVICE

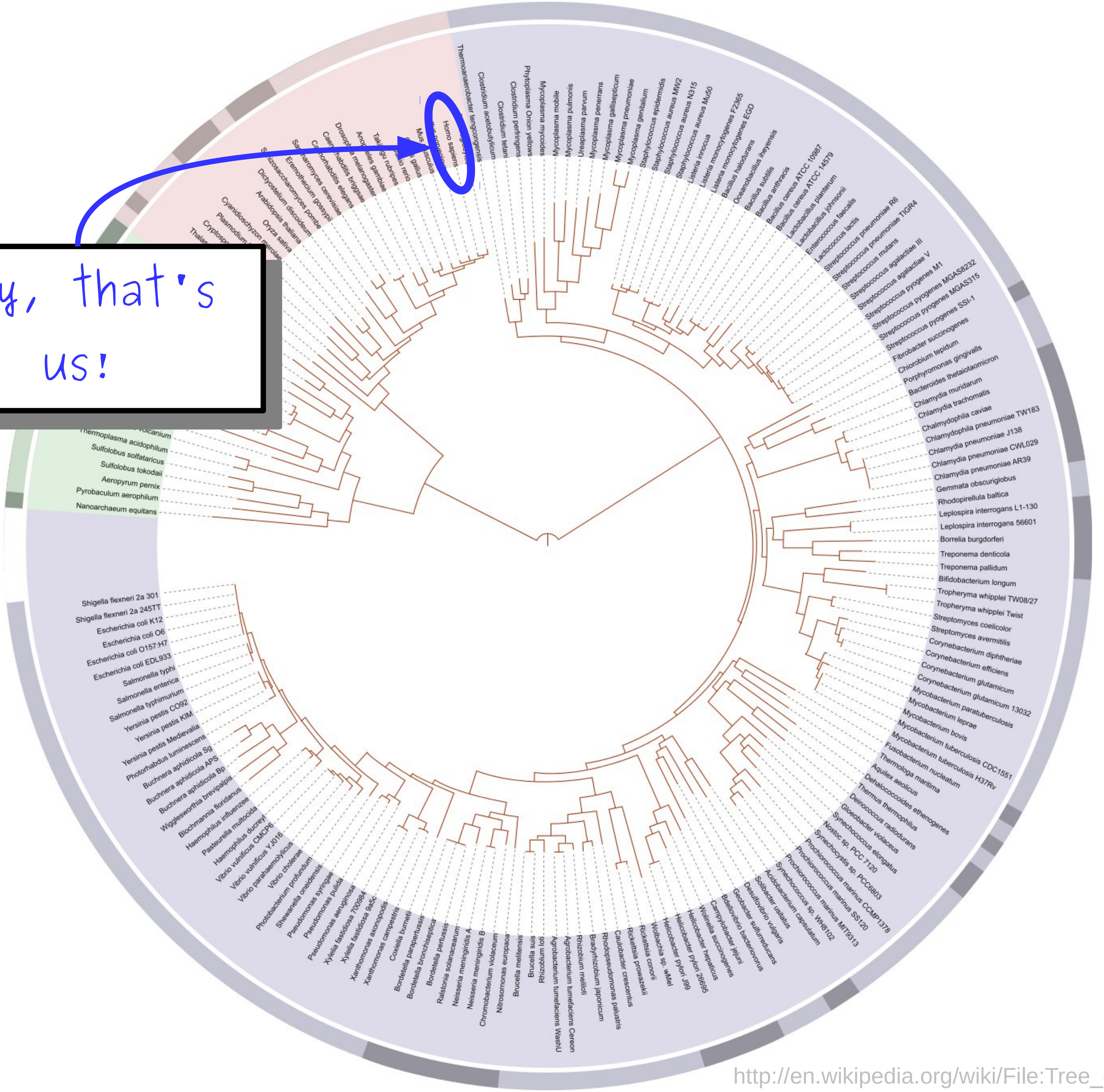
http://www.sawyoo.com/postpic/2014/11/executive-branch-u-s-government-chart_616676.jpg





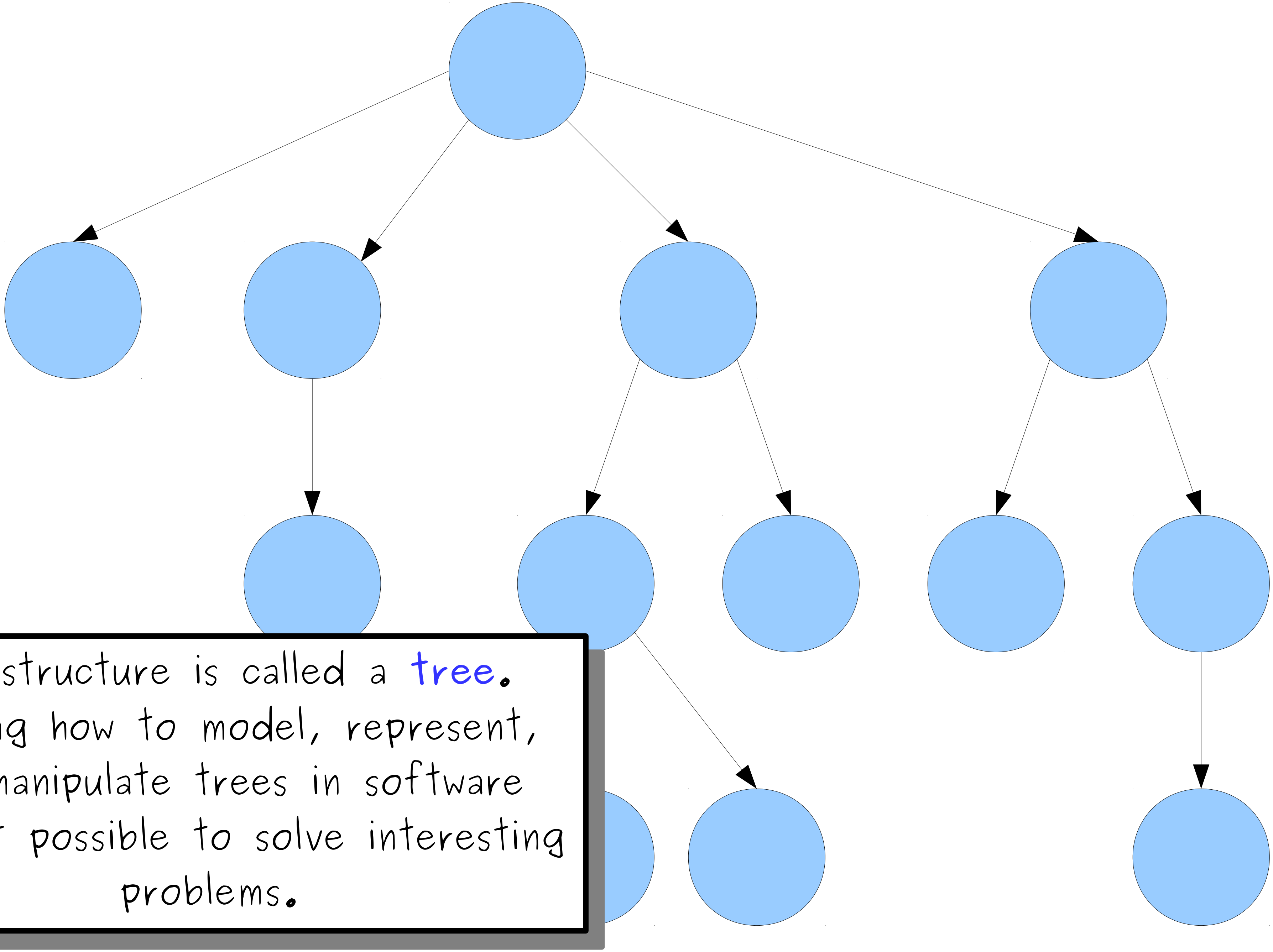






Hey, that's us!

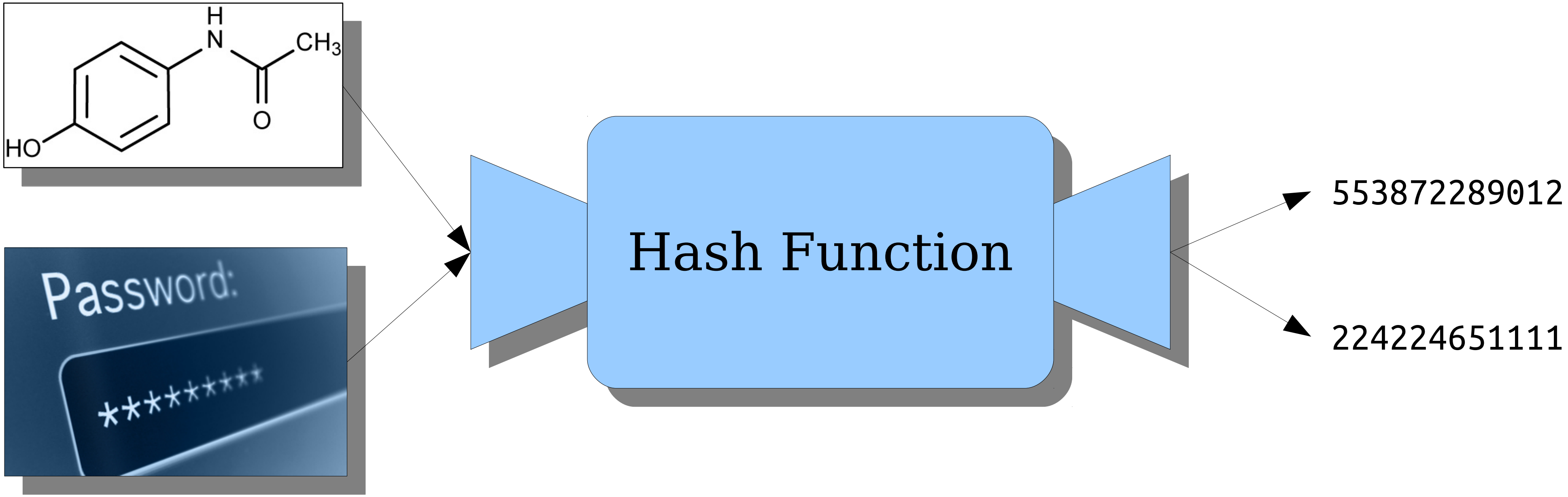
http://en.wikipedia.org/wiki/File:Tree_of_life_SVG.svg



This structure is called a **tree**. Knowing how to model, represent, and manipulate trees in software makes it possible to solve interesting problems.

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

How do we keep passwords secure when servers are hacked all the time?



Hash Function

553872289012

224224651111

Inputs can be just about anything: strings, ID numbers, molecular shapes, passwords, etc.

Output is a seemingly random number that serves as a "fingerprint" of the input.

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

Goals for this 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.
  - Quantitatively analyze different approaches for solving problems.

Goals for this Course

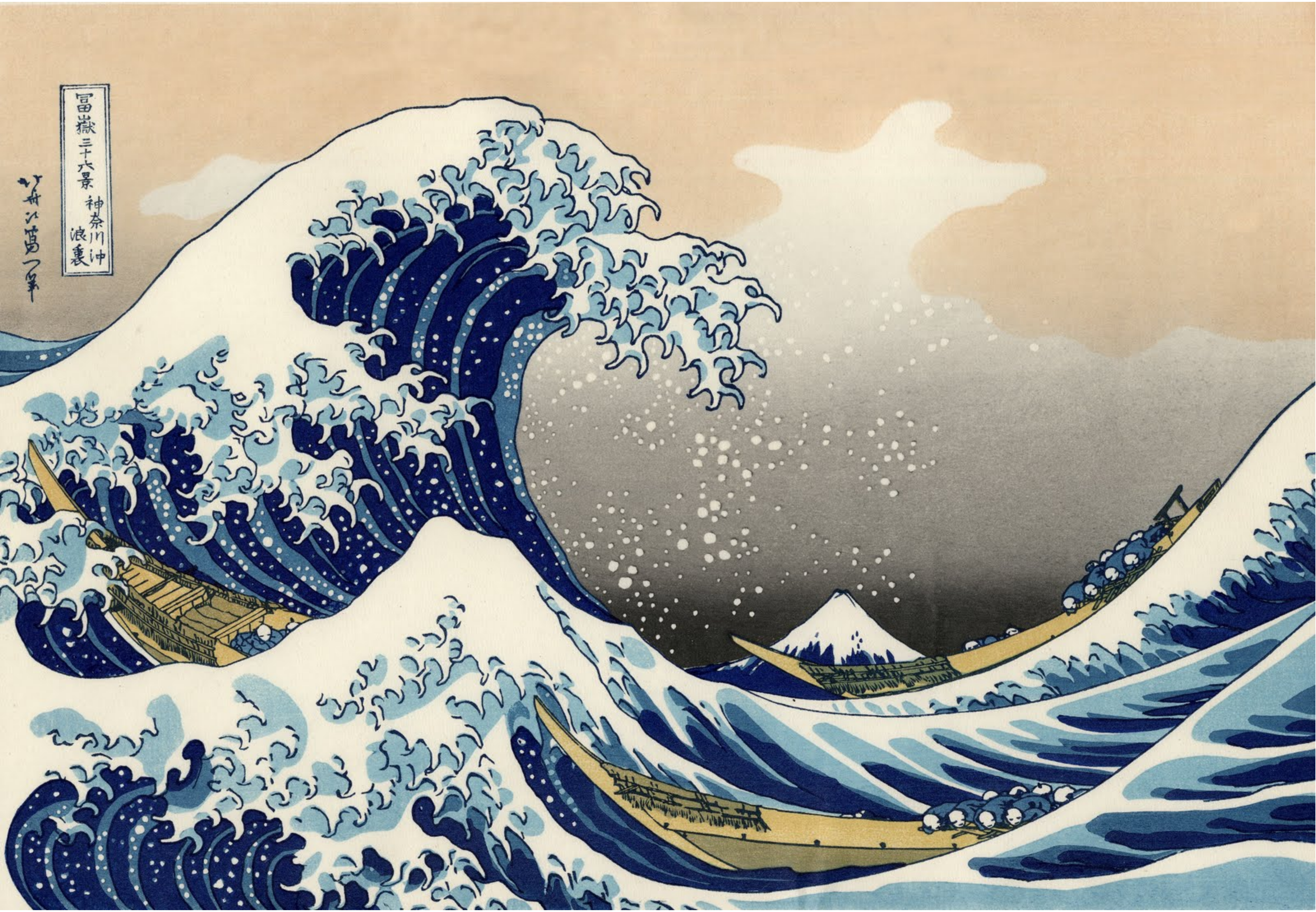
Learn how to model and solve complex problems with computers.

To that end:

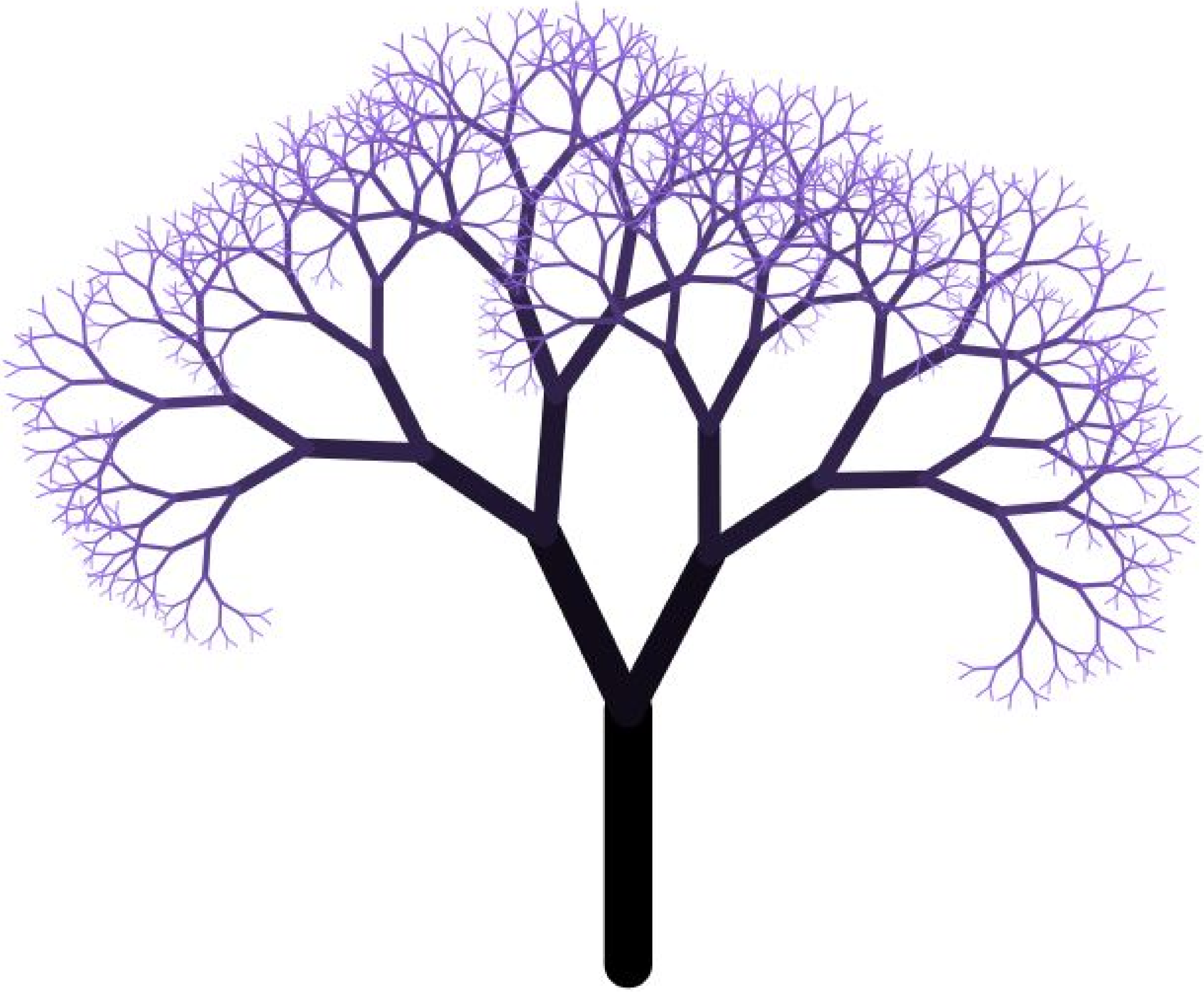
Explore common abstractions for representing problems.

- Harness recursion and understand how to think about problems recursively.

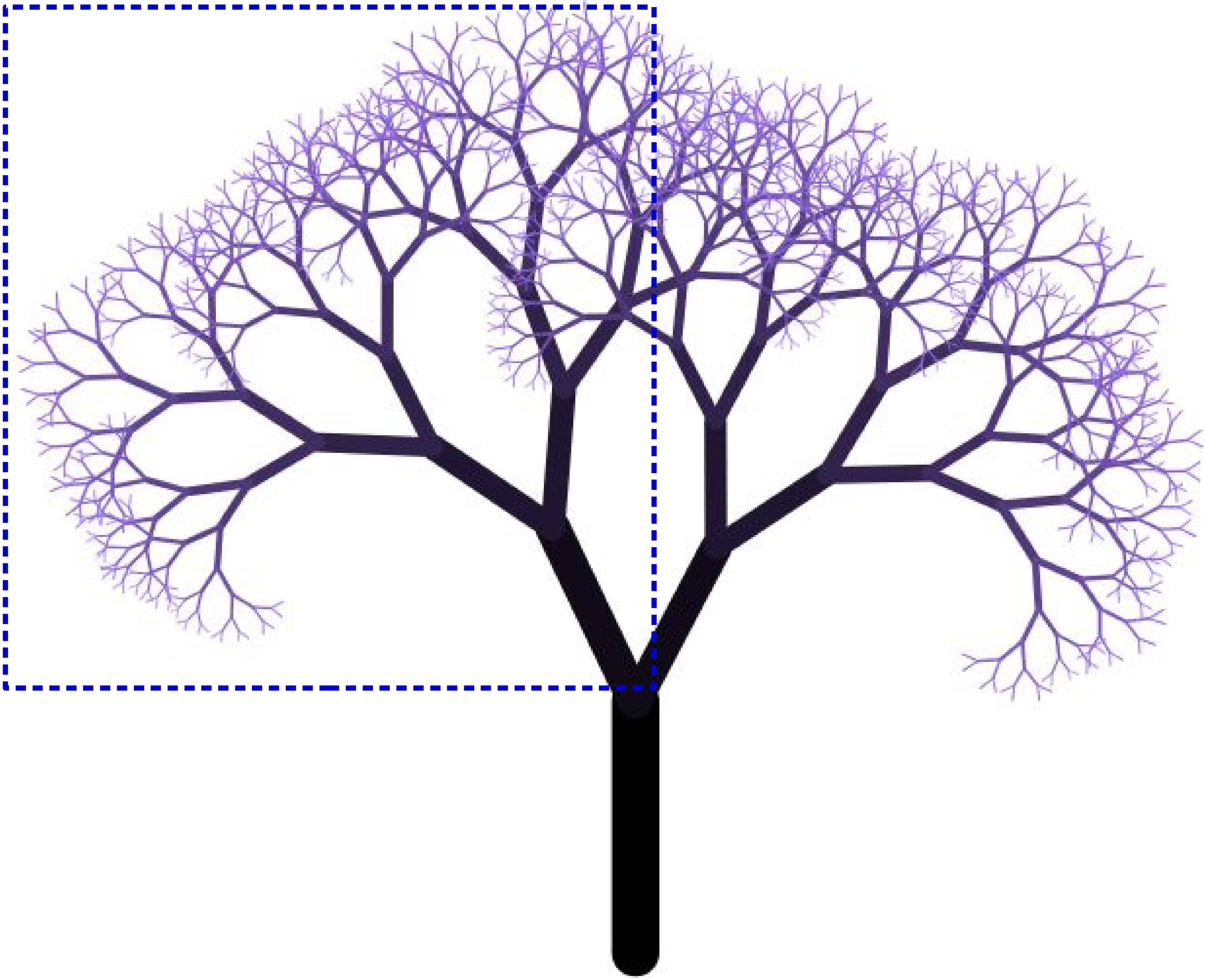
Quantitatively analyze different approaches for solving problems.



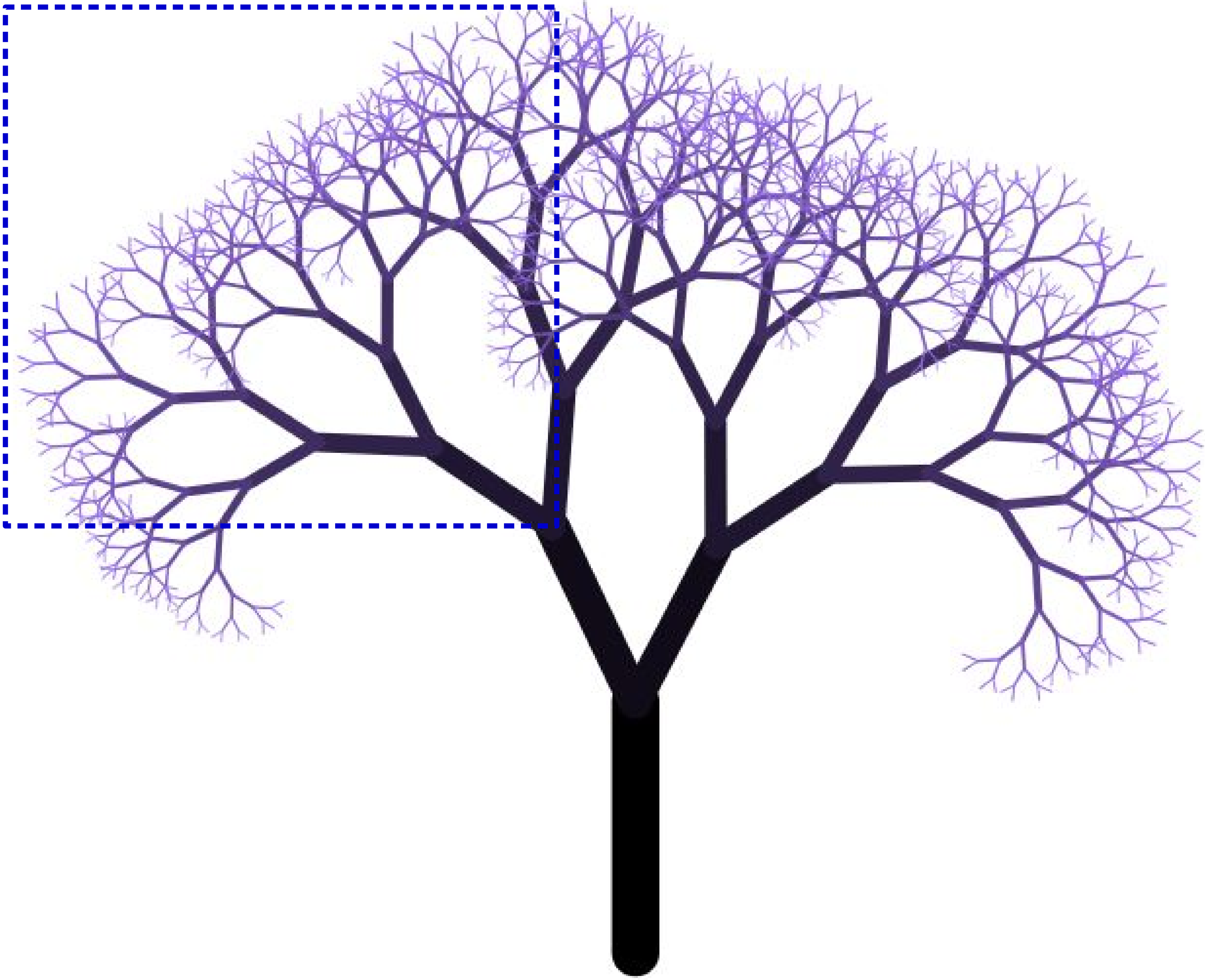
http://1.bp.blogspot.com/-Hwf-U9hRwEI/TX480ONoo6I/AAAAAAAAAcM/fyBi0I5HsUI/s1600/the_great_wave_off_kanagawa.jpg



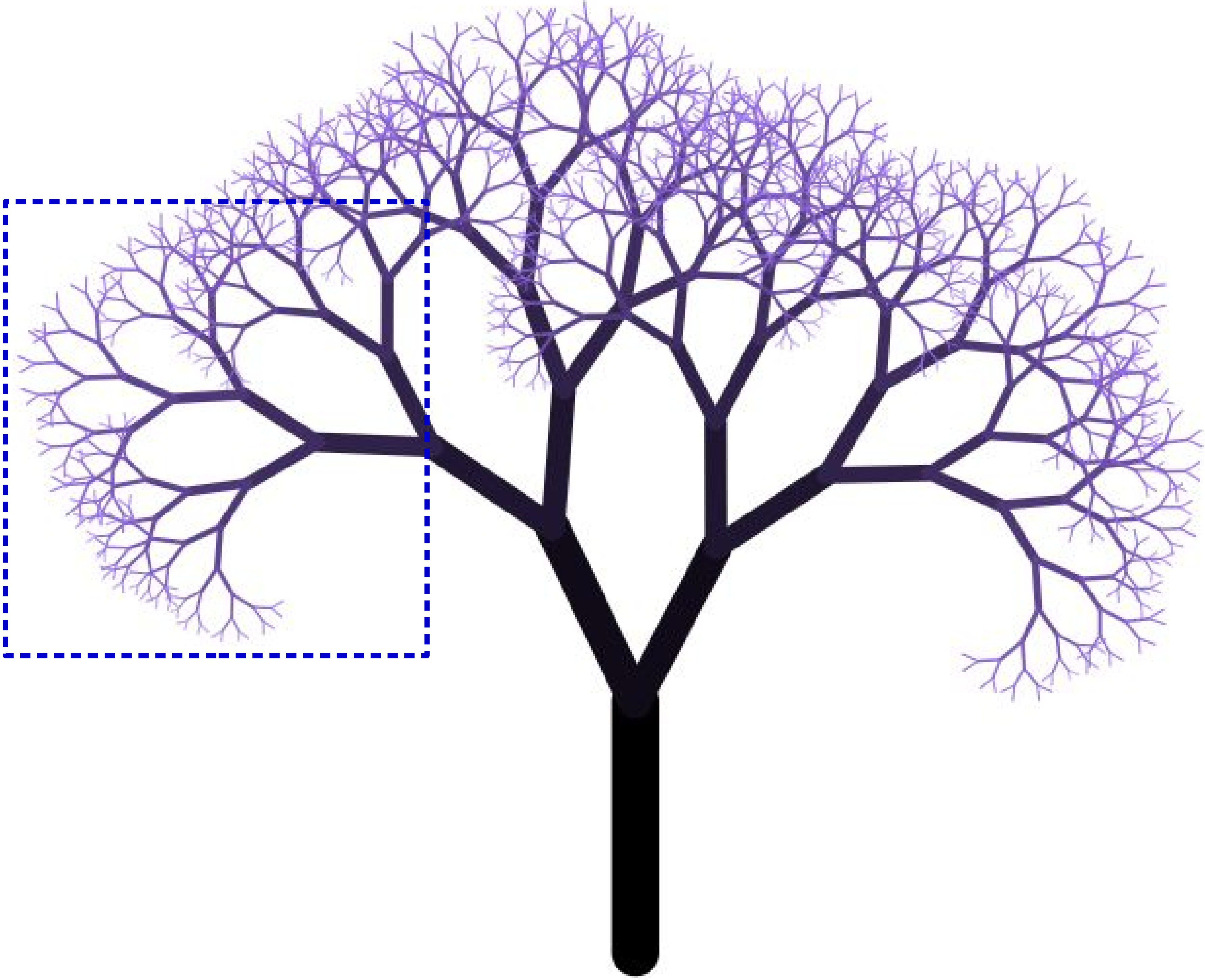
http://www.marketoracle.co.uk/images/2010/Oct/fractal-tree2.jpg



http://www.marketoracle.co.uk/images/2010/Oct/fractal-tree2.jpg



http://www.marketoracle.co.uk/images/2010/Oct/fractal-tree2.jpg



http://www.marketoracle.co.uk/images/2010/Oct/fractal-tree2.jpg

Creating Trees

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

Thinking recursively gives you a different perspective on everyday structures.

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

Goals for this 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.
  - Quantitatively analyze different approaches for solving problems.

Goals for this Course

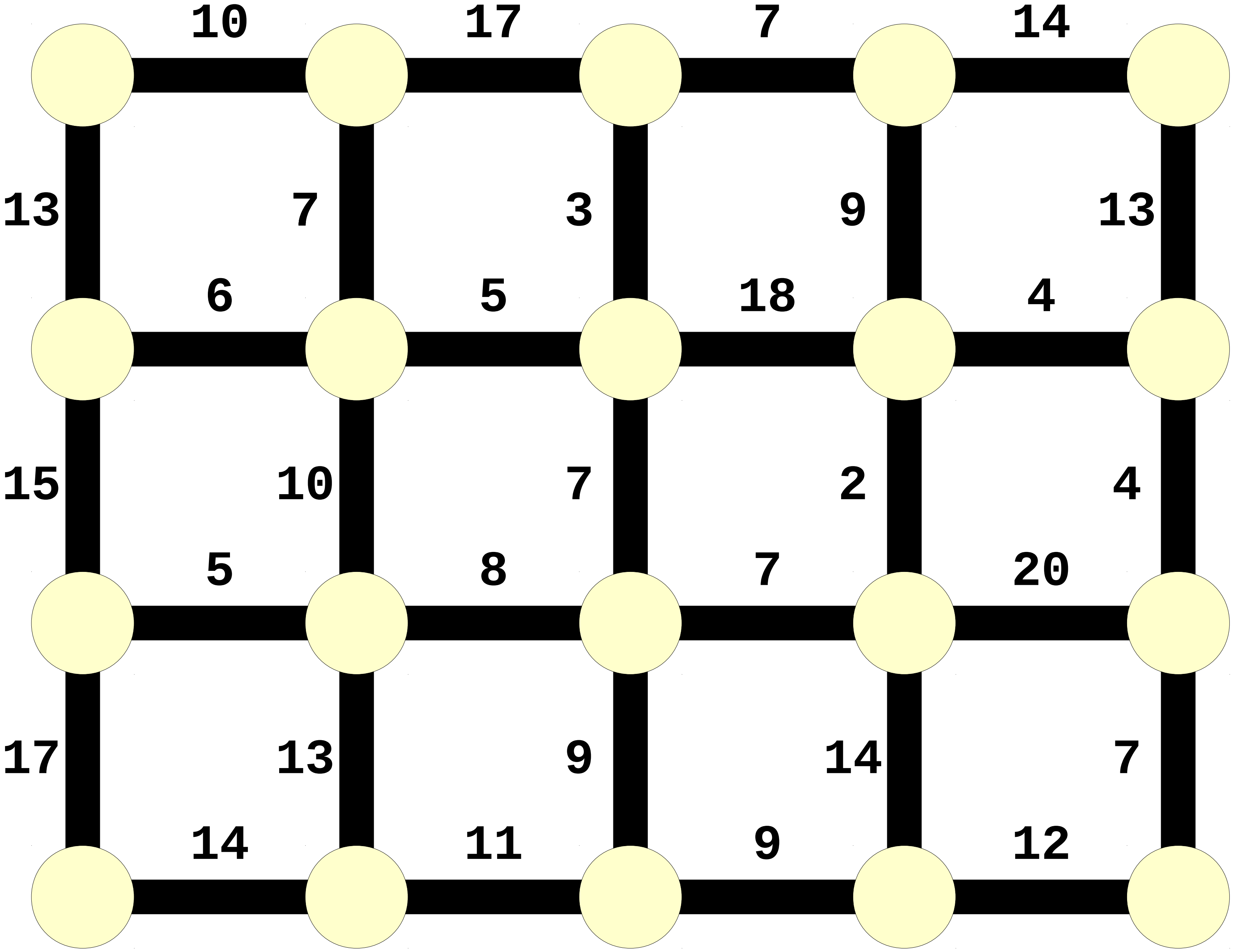
Learn how to model and solve complex problems with computers.

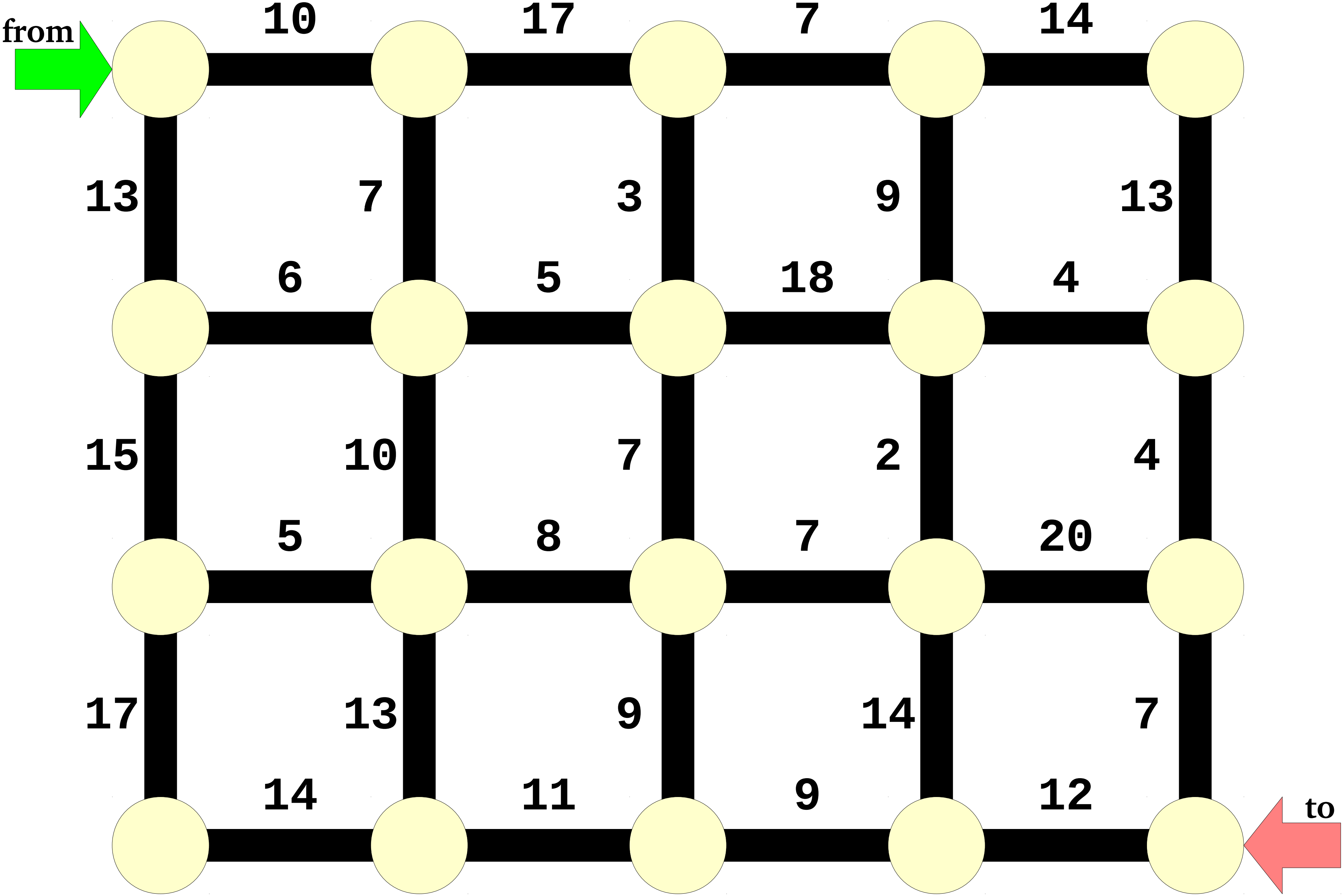
To that end:

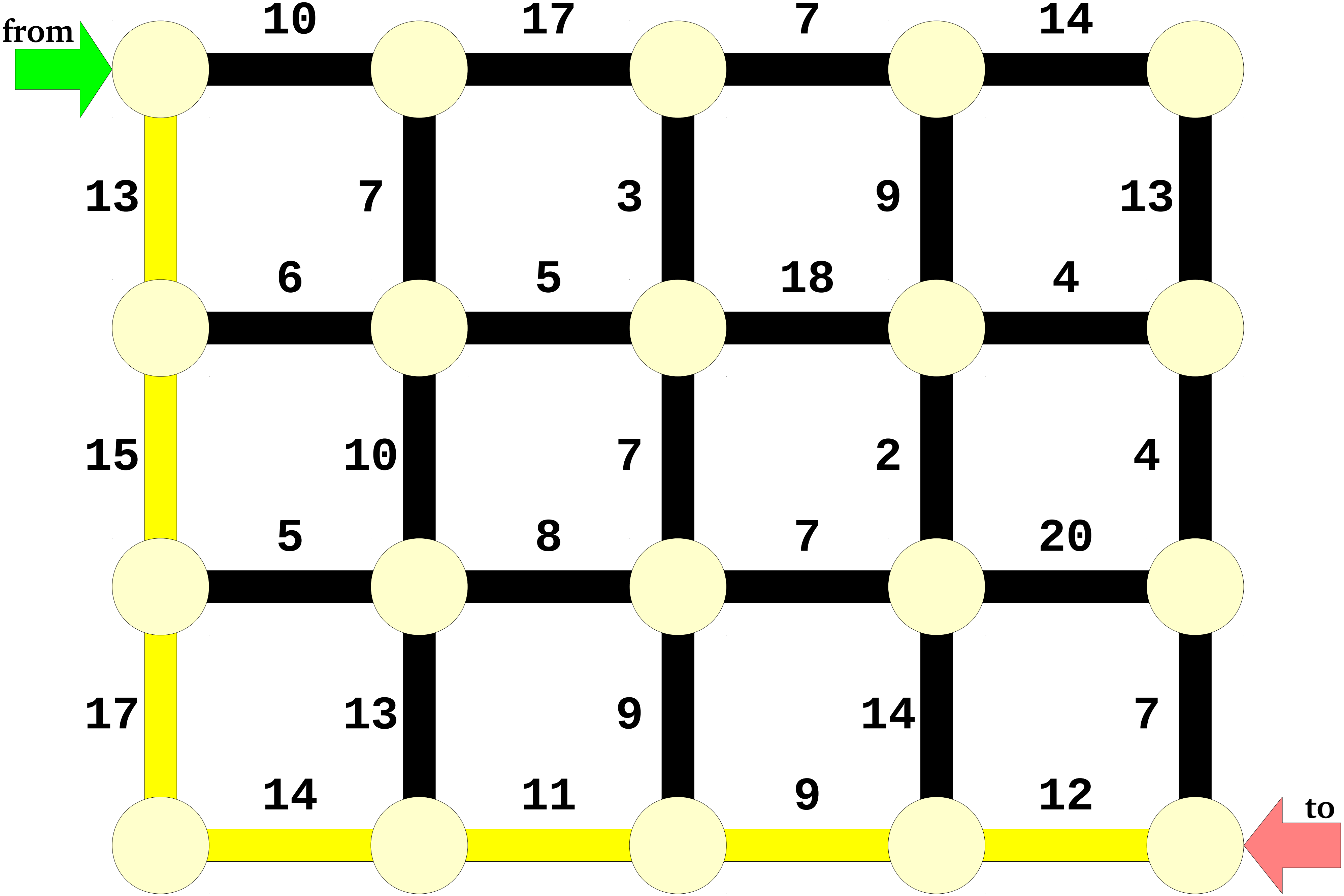
Explore common abstractions for representing problems.

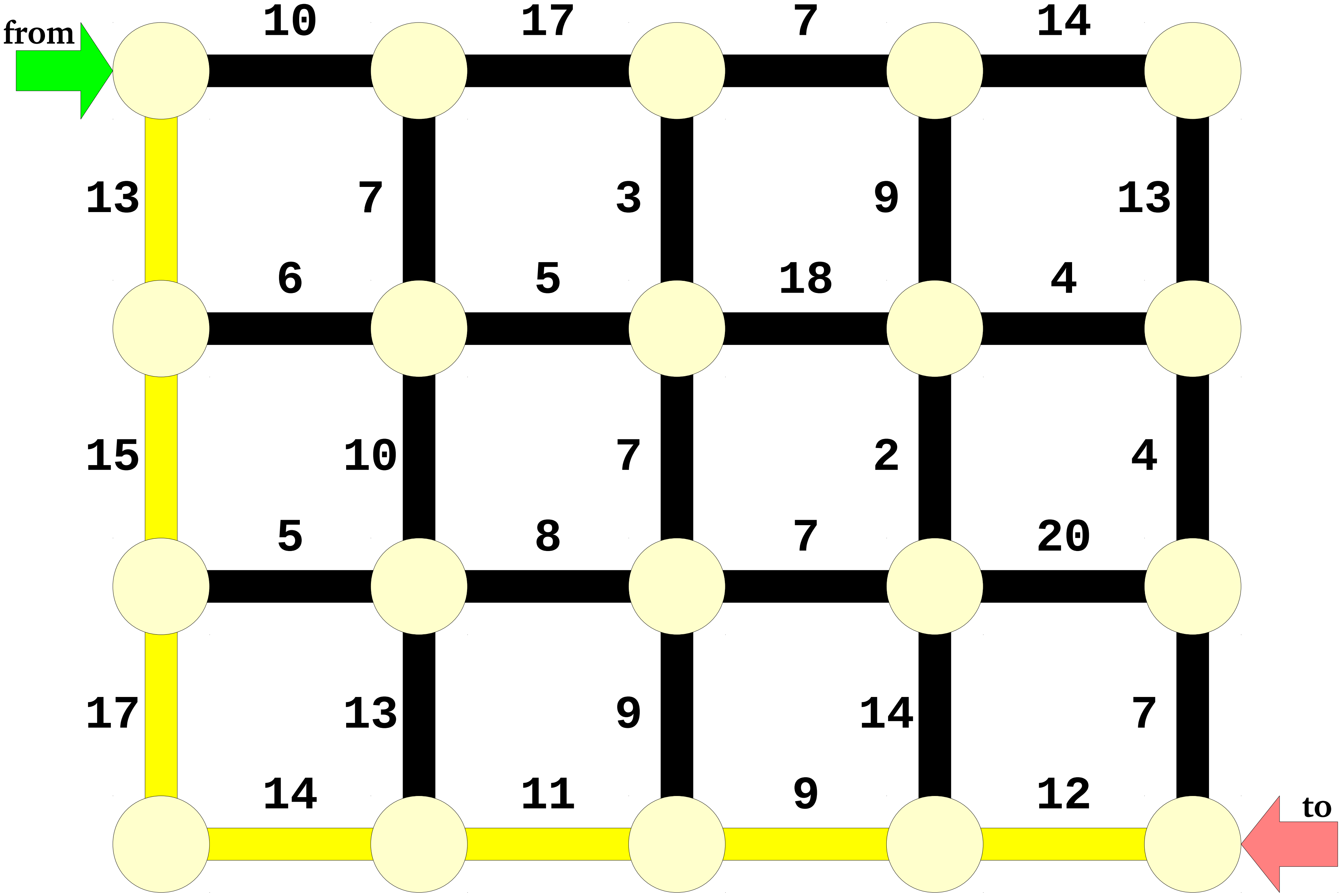
Harness recursion and understand how to think about problems recursively.

- Quantitatively analyze different approaches for solving problems.

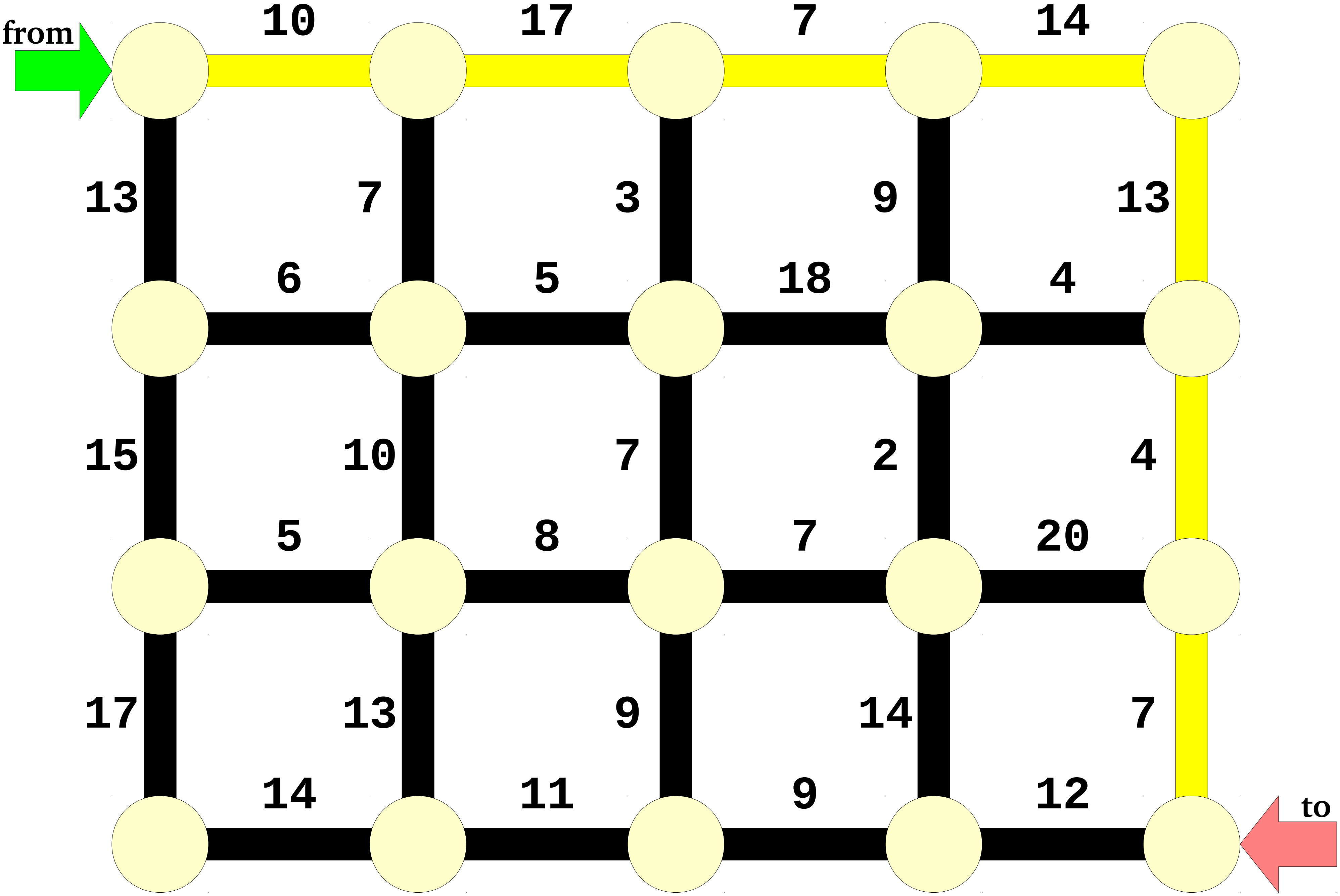

10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12

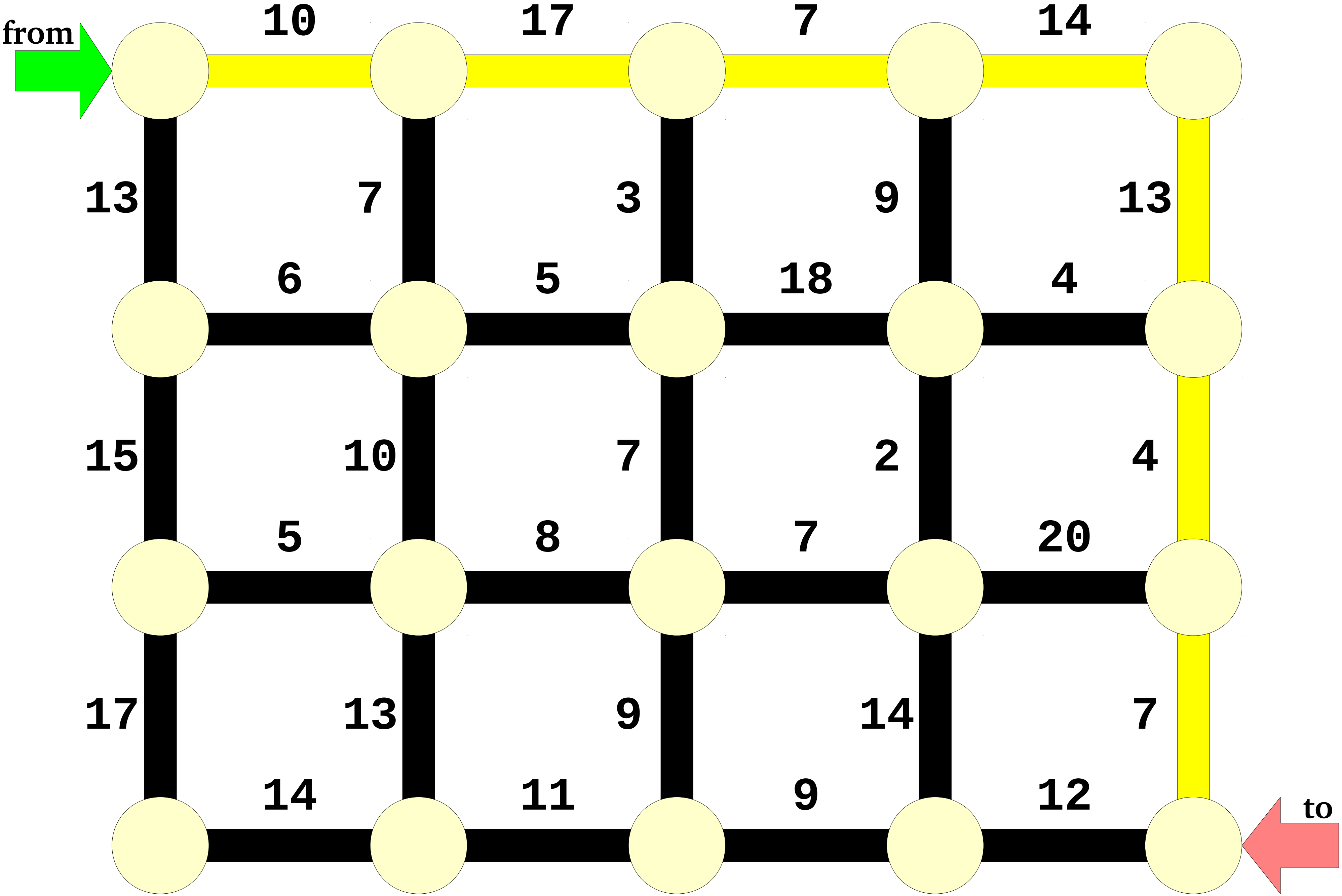

from
10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to


from
10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

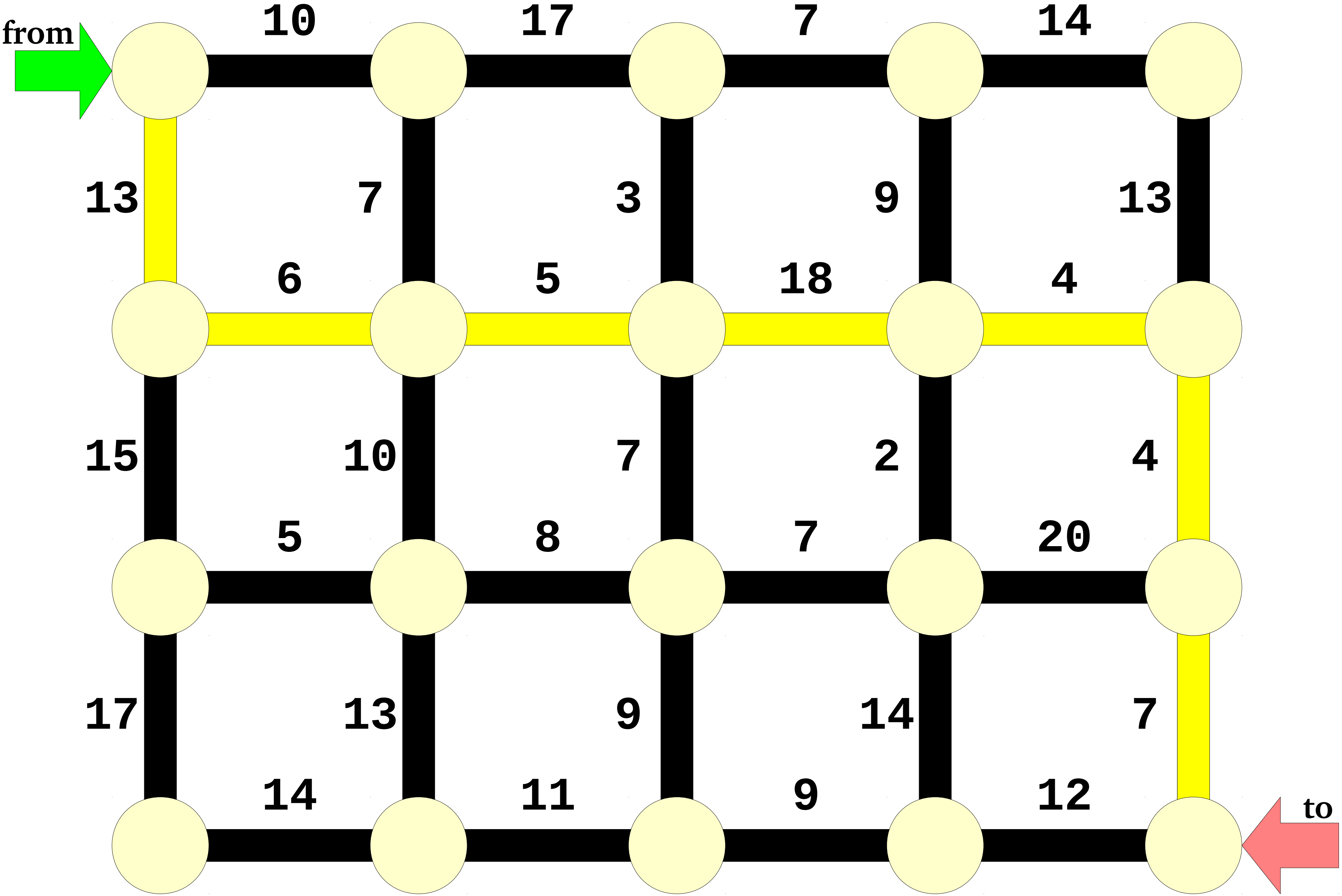


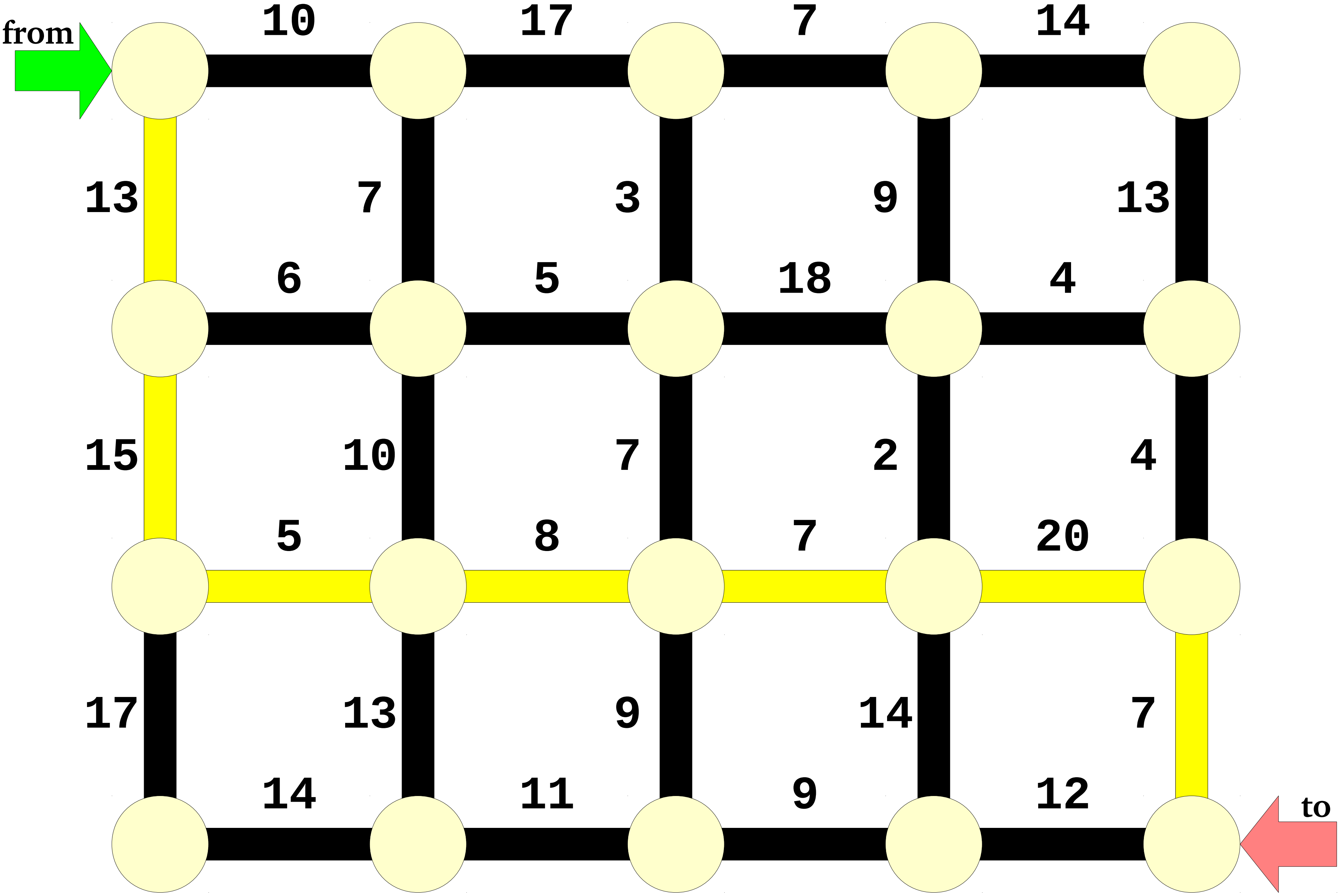
Travel Time: 13 + 15 + 17 + 14 + 11 + 9 + 12 = **91**

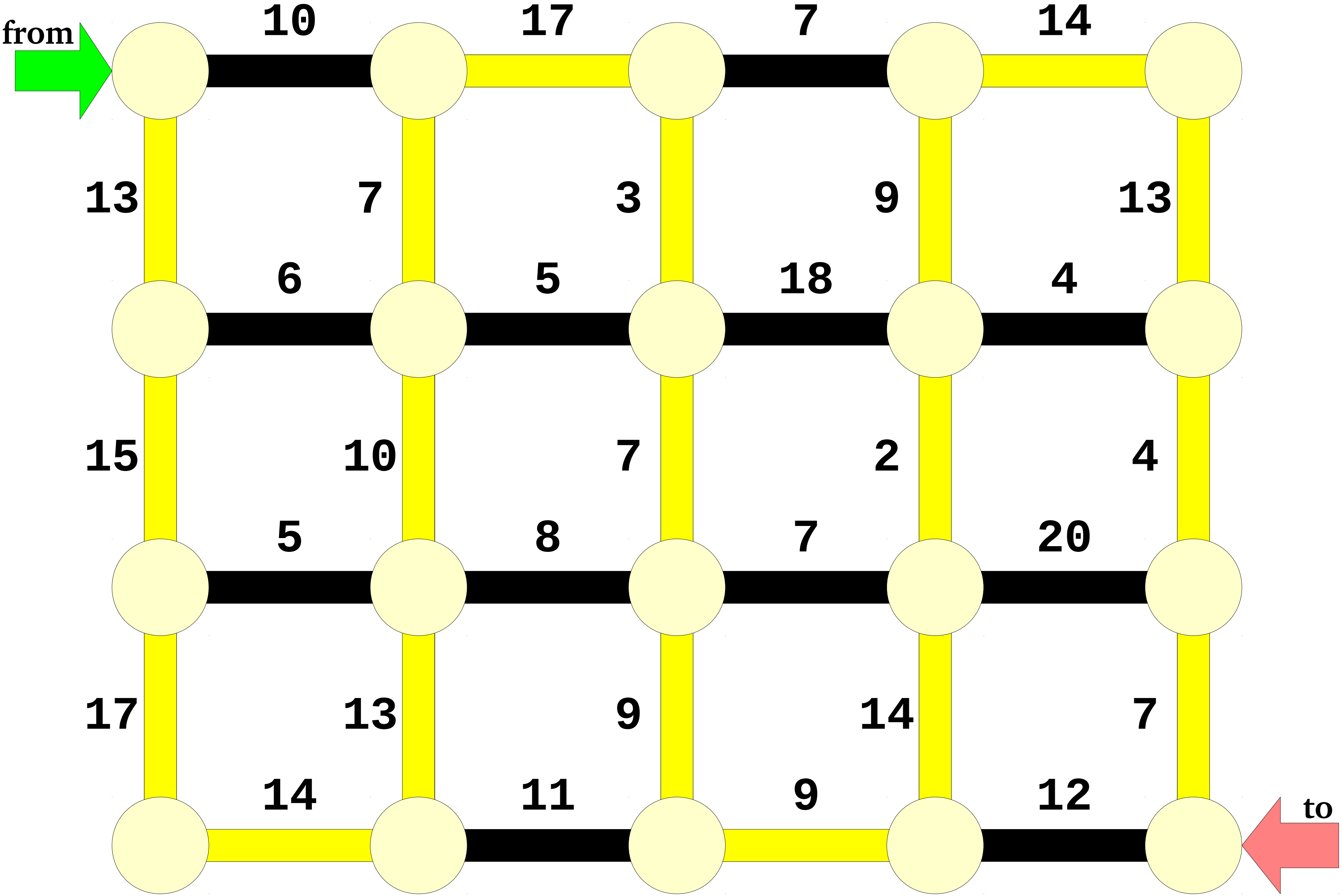

from
10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

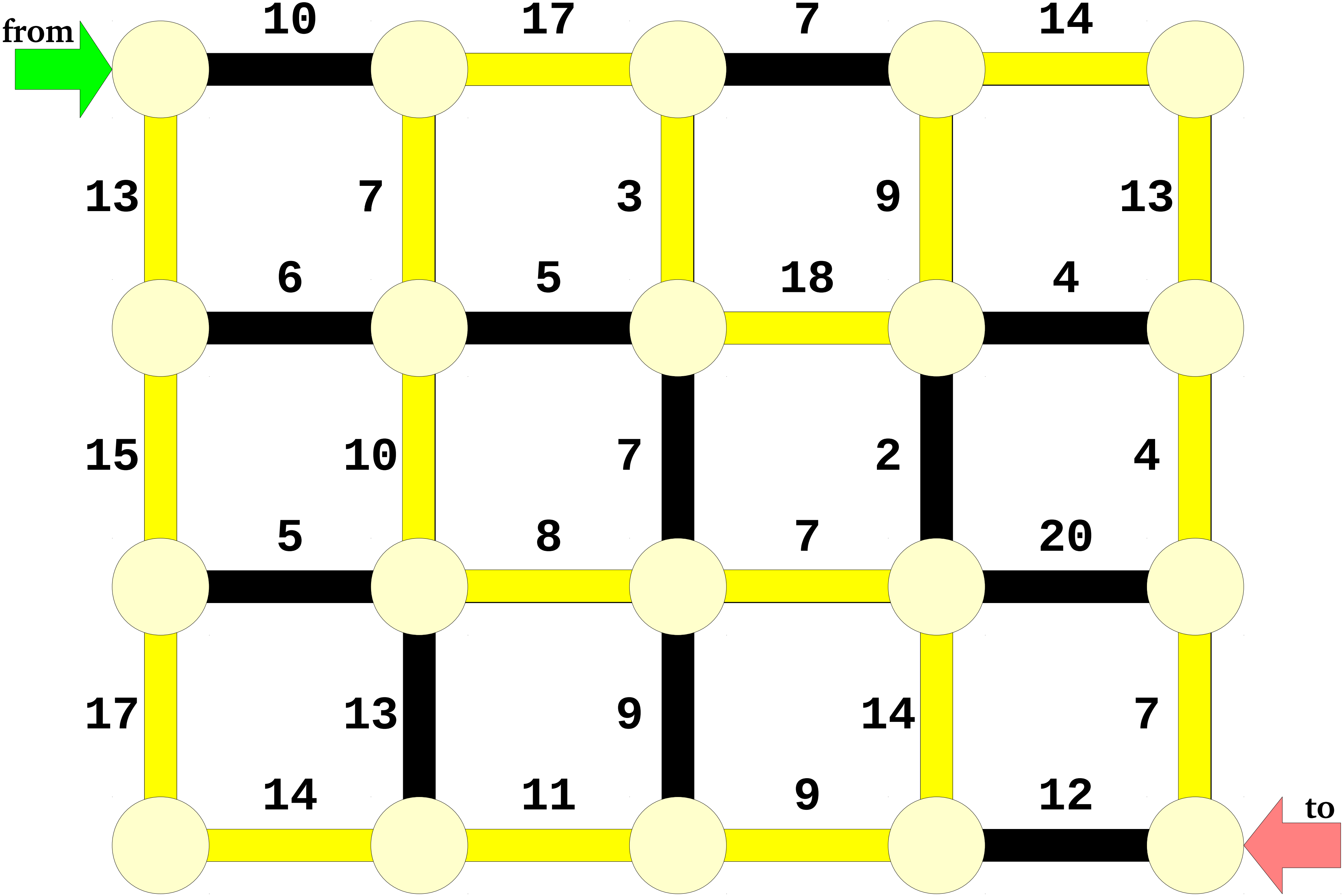


Travel Time: 10 + 17 + 7 + 14 + 13 + 4 + 7 = **72**

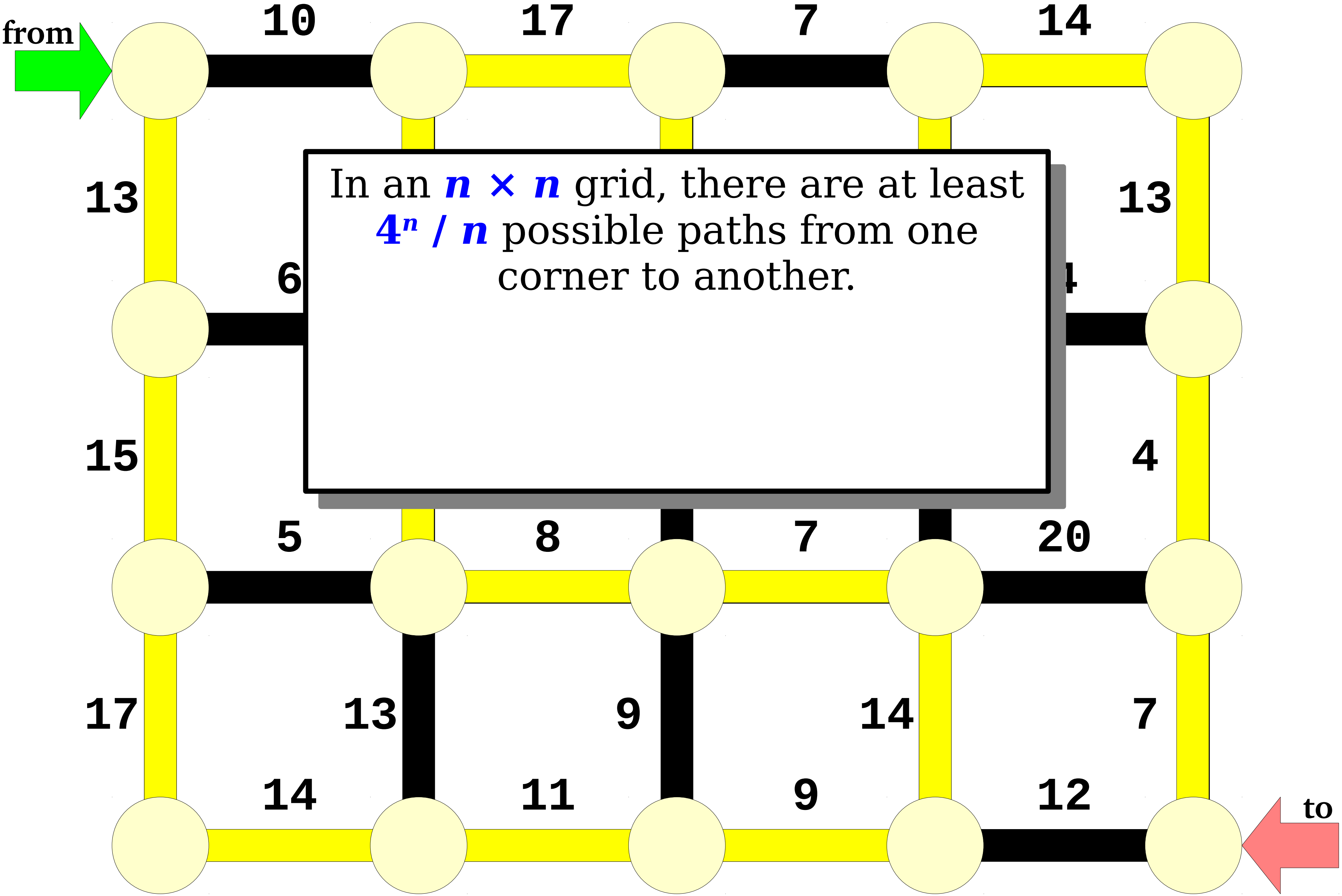

from
10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to


from
10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

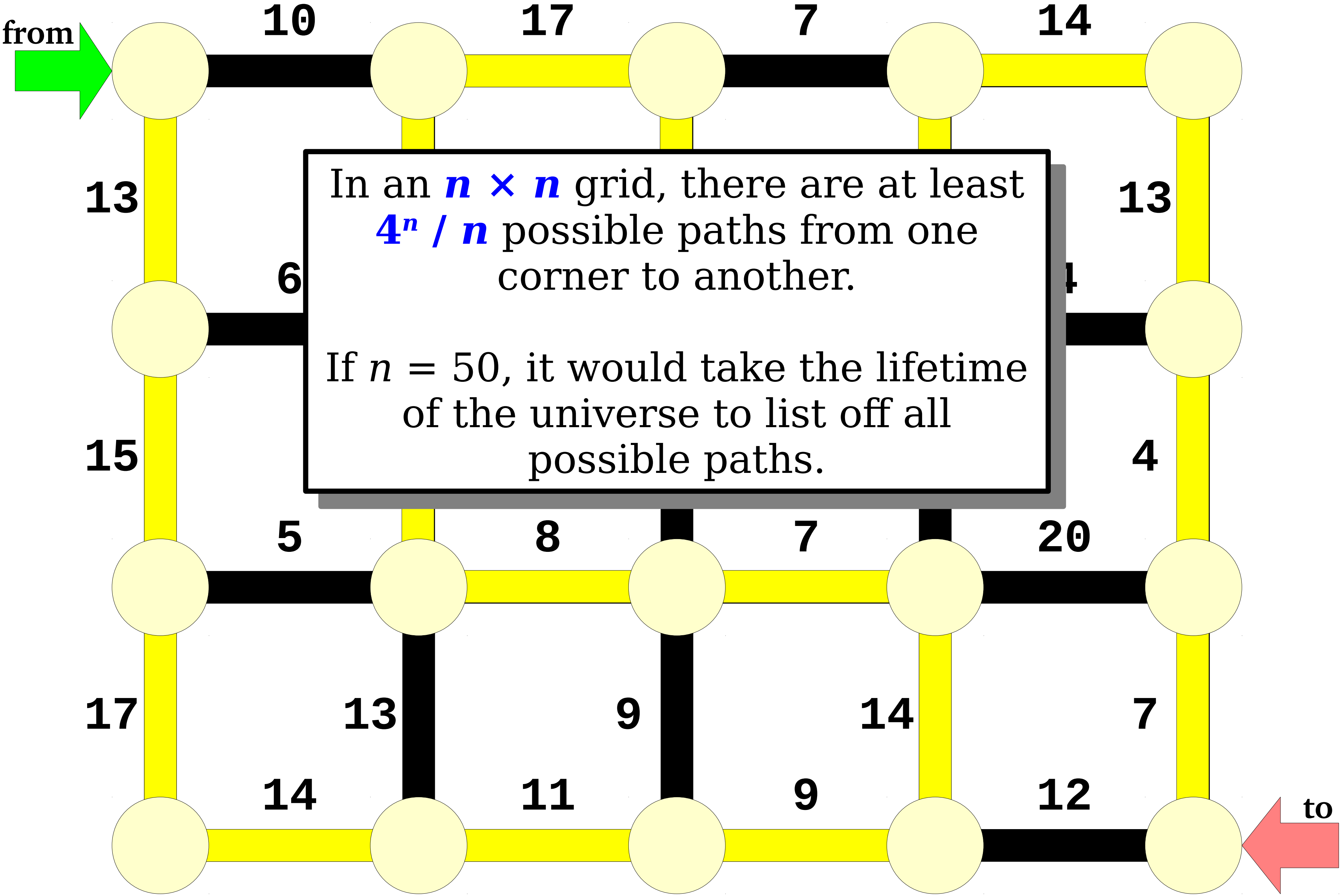

from
10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to



from
10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

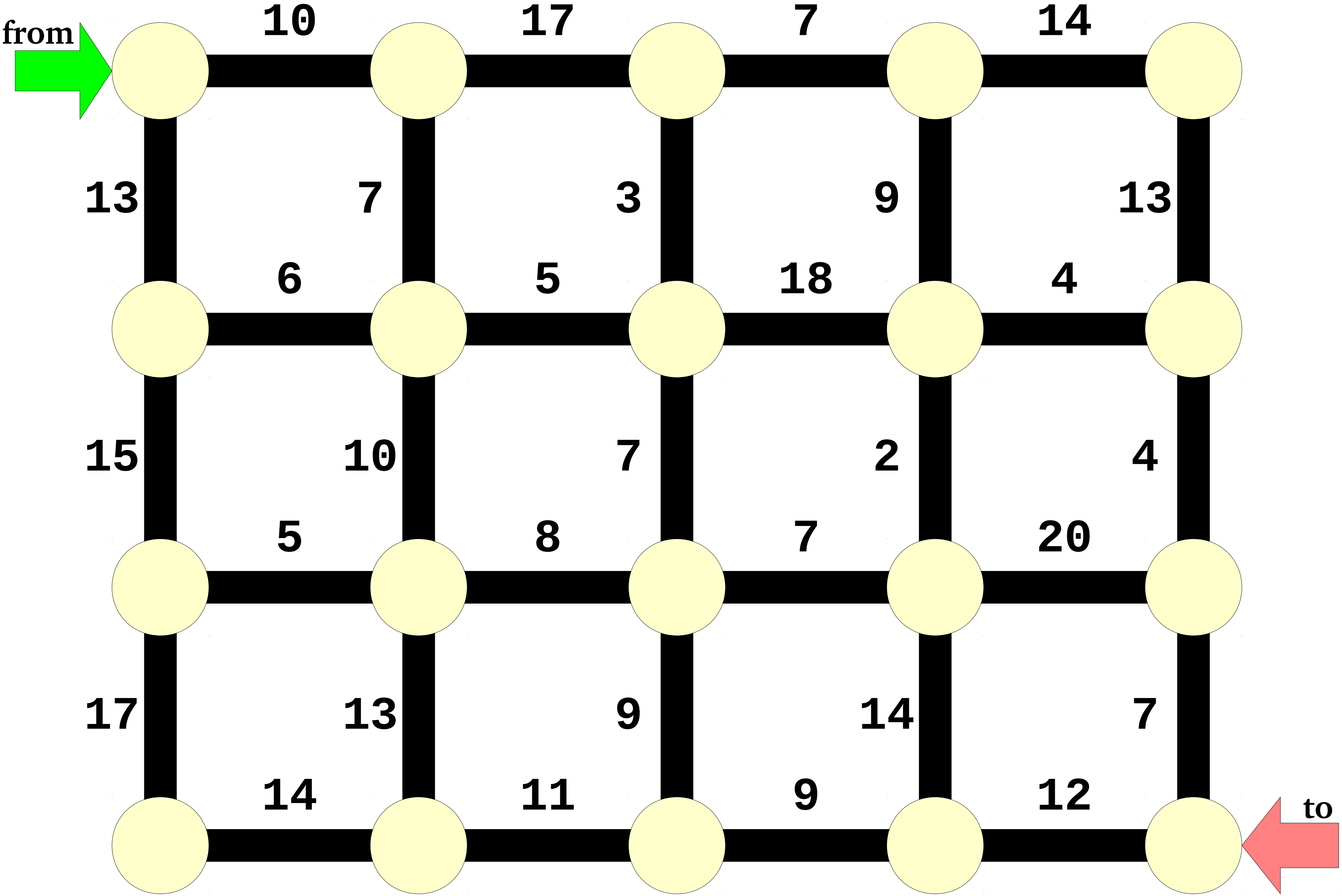


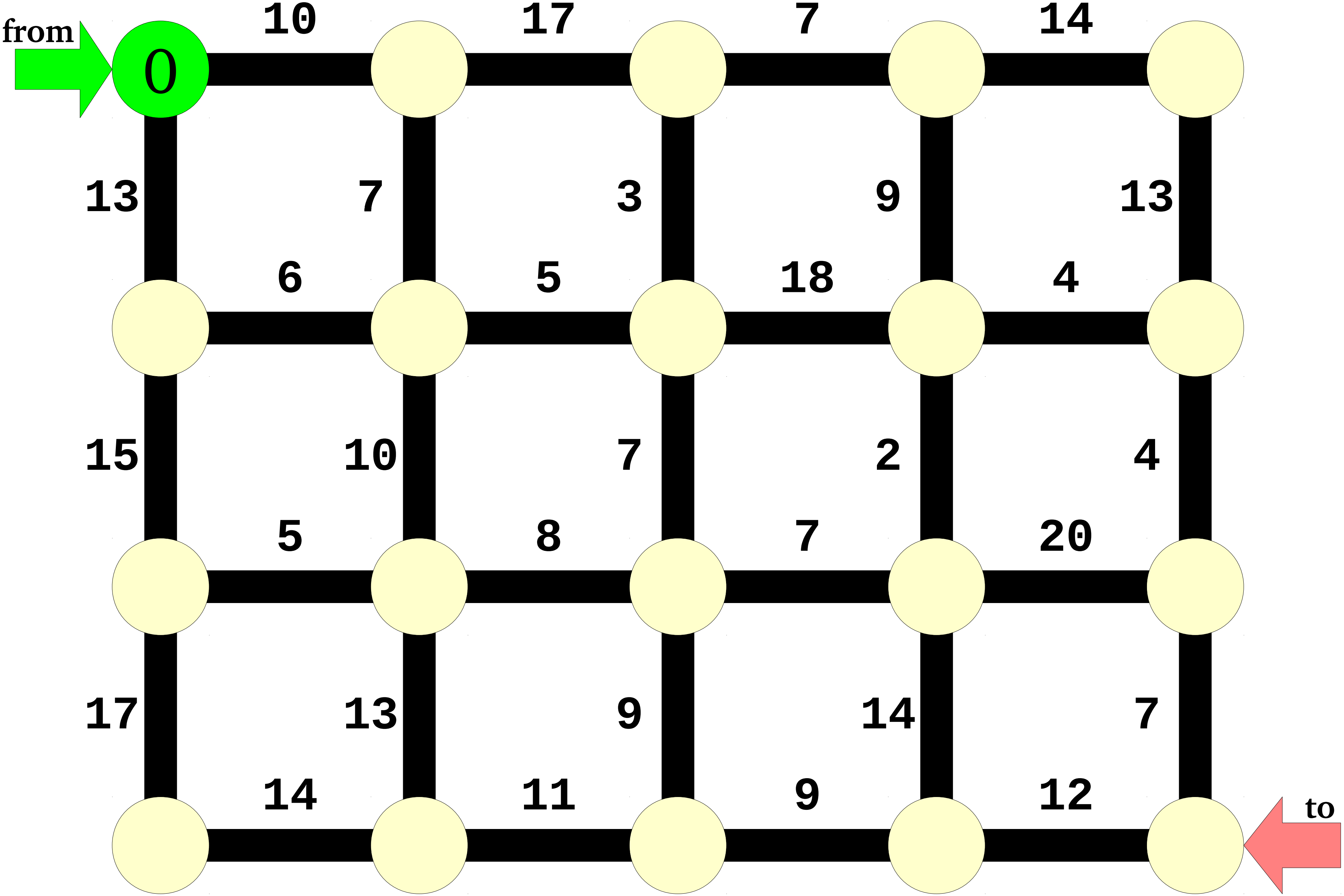
In an $n \times n$ grid, there are at least $4^n / n$ possible paths from one corner to another.

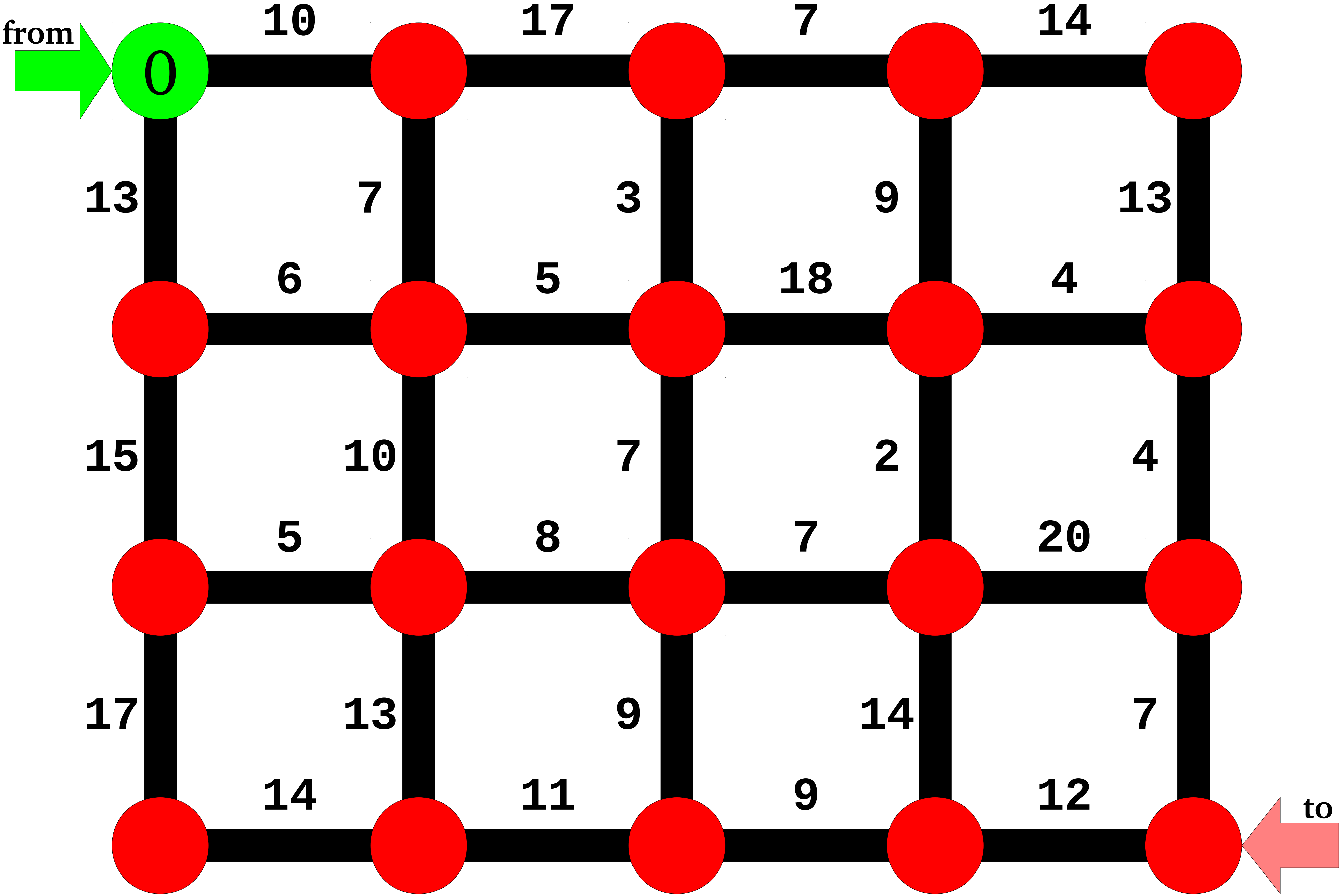


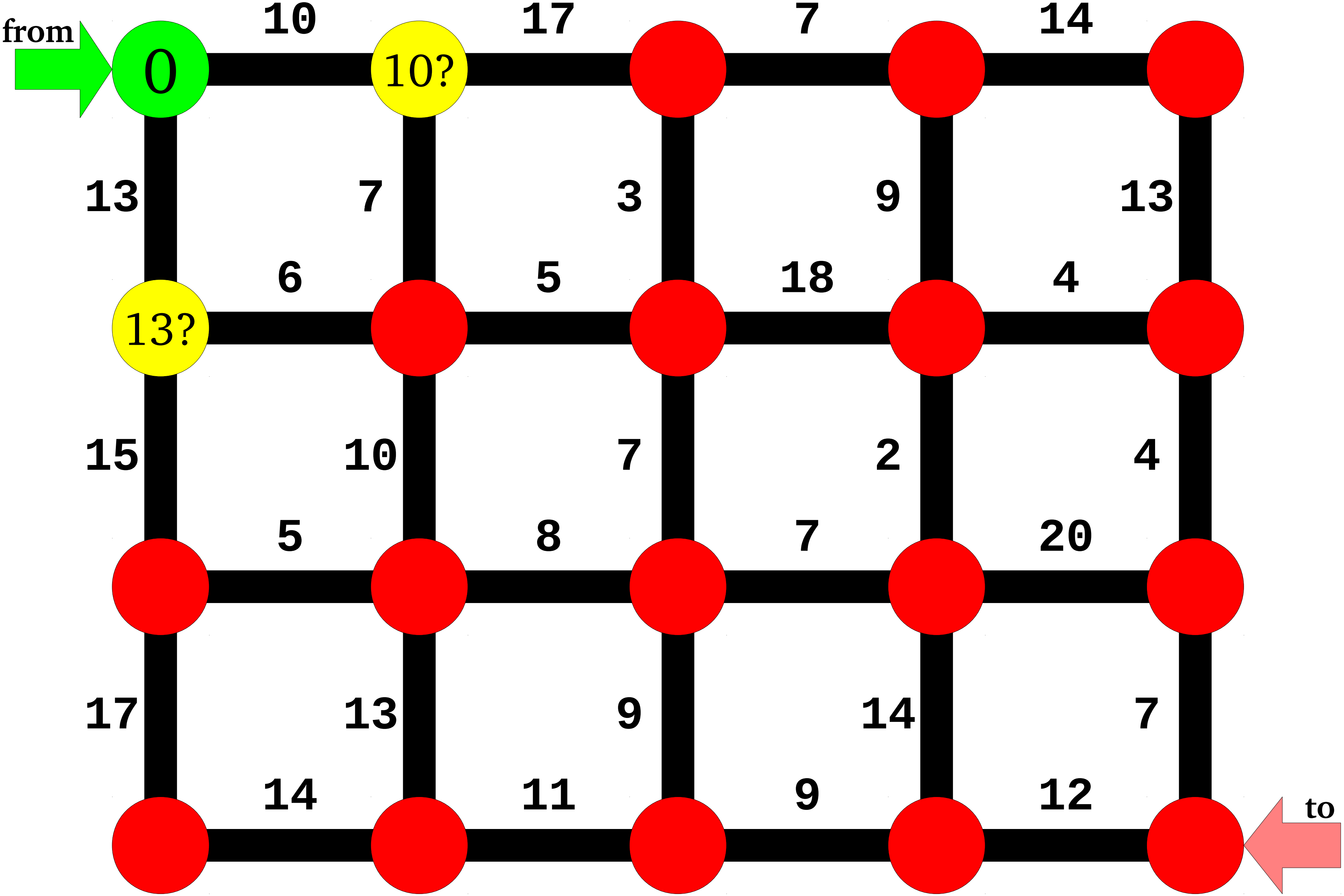
In an $n \times n$ grid, there are at least $4^n / n$ possible paths from one corner to another.

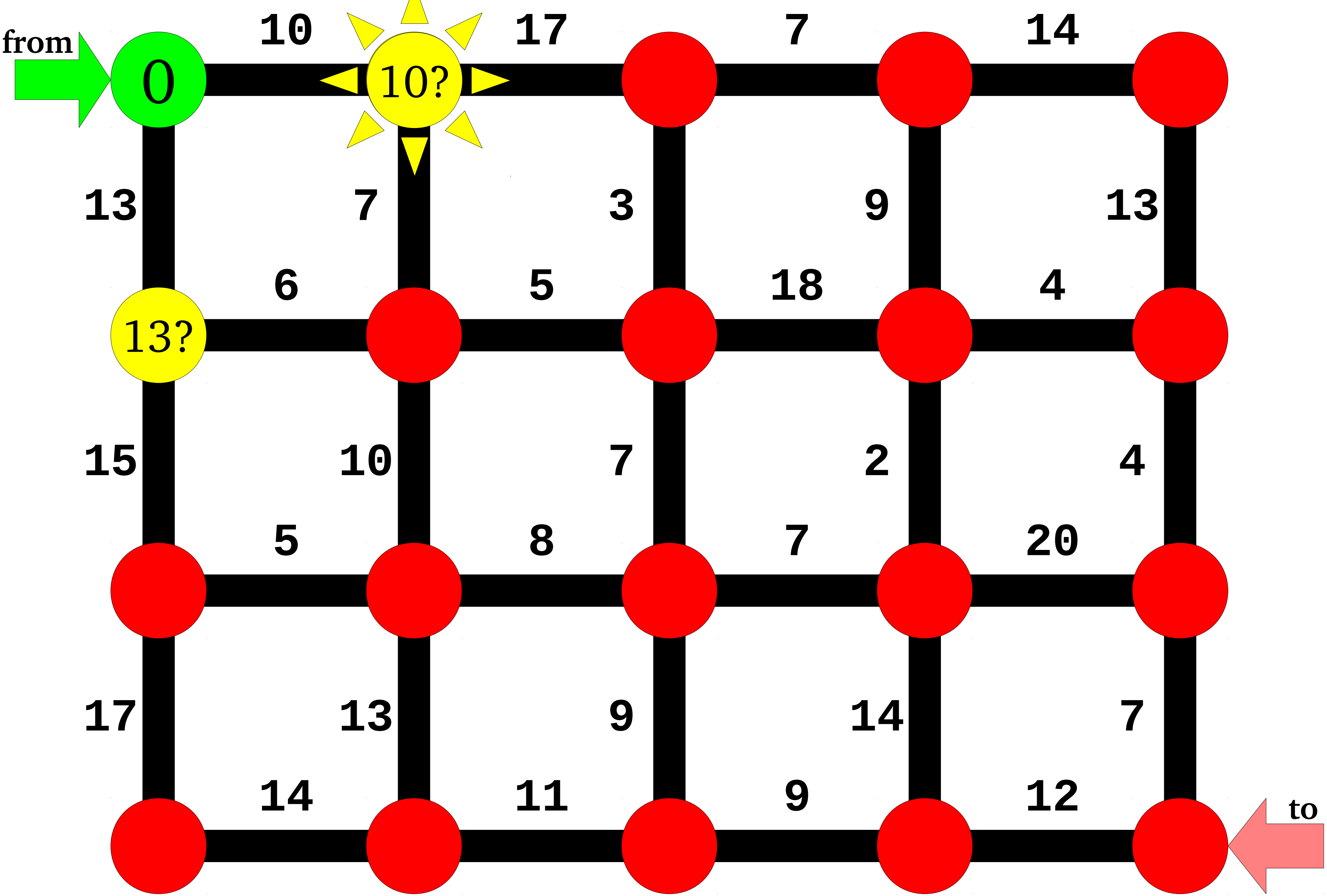
If n = 50, it would take the lifetime of the universe to list off all possible paths.

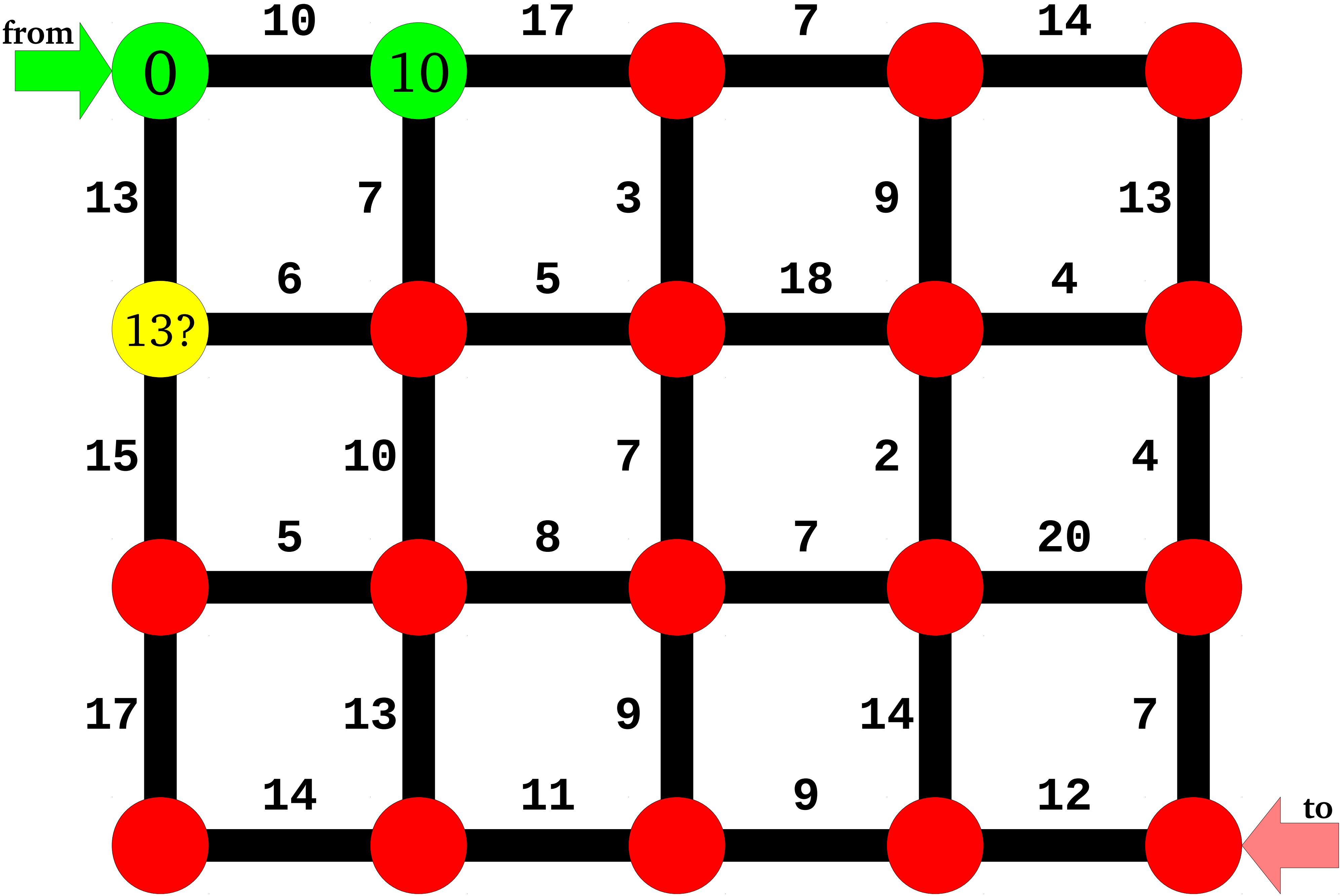

from
10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

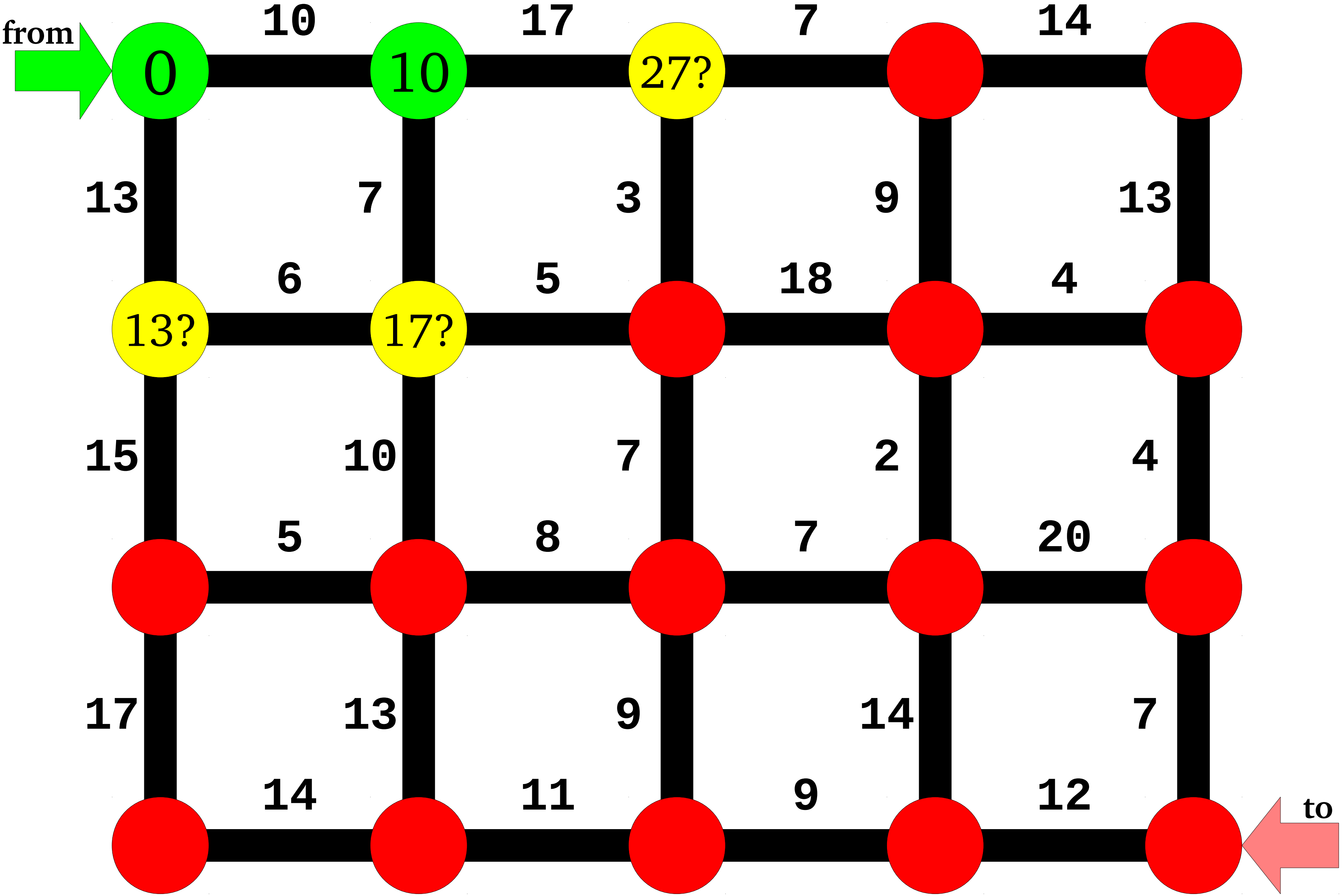

from
0
10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

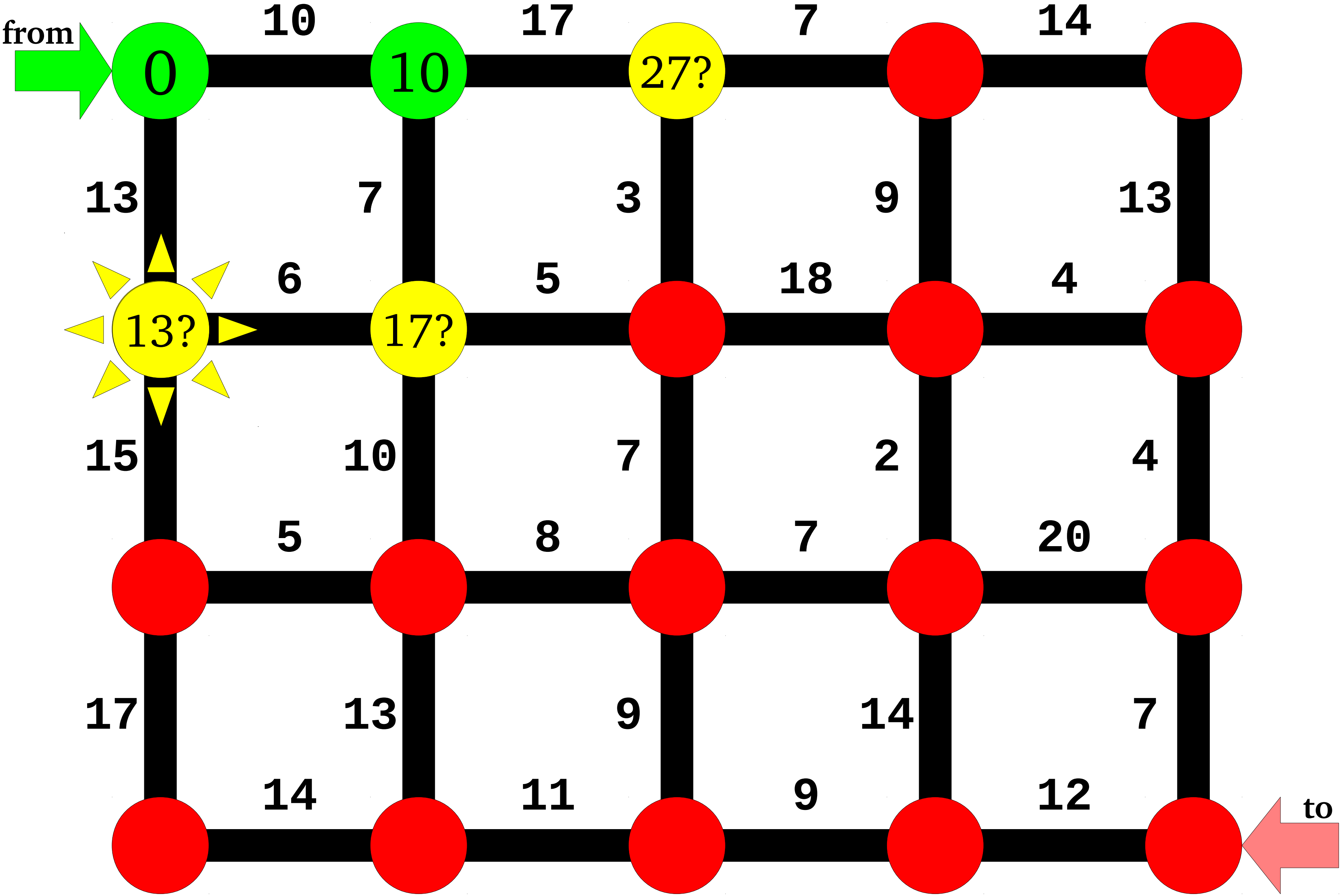

from
10
17
7
14
0
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

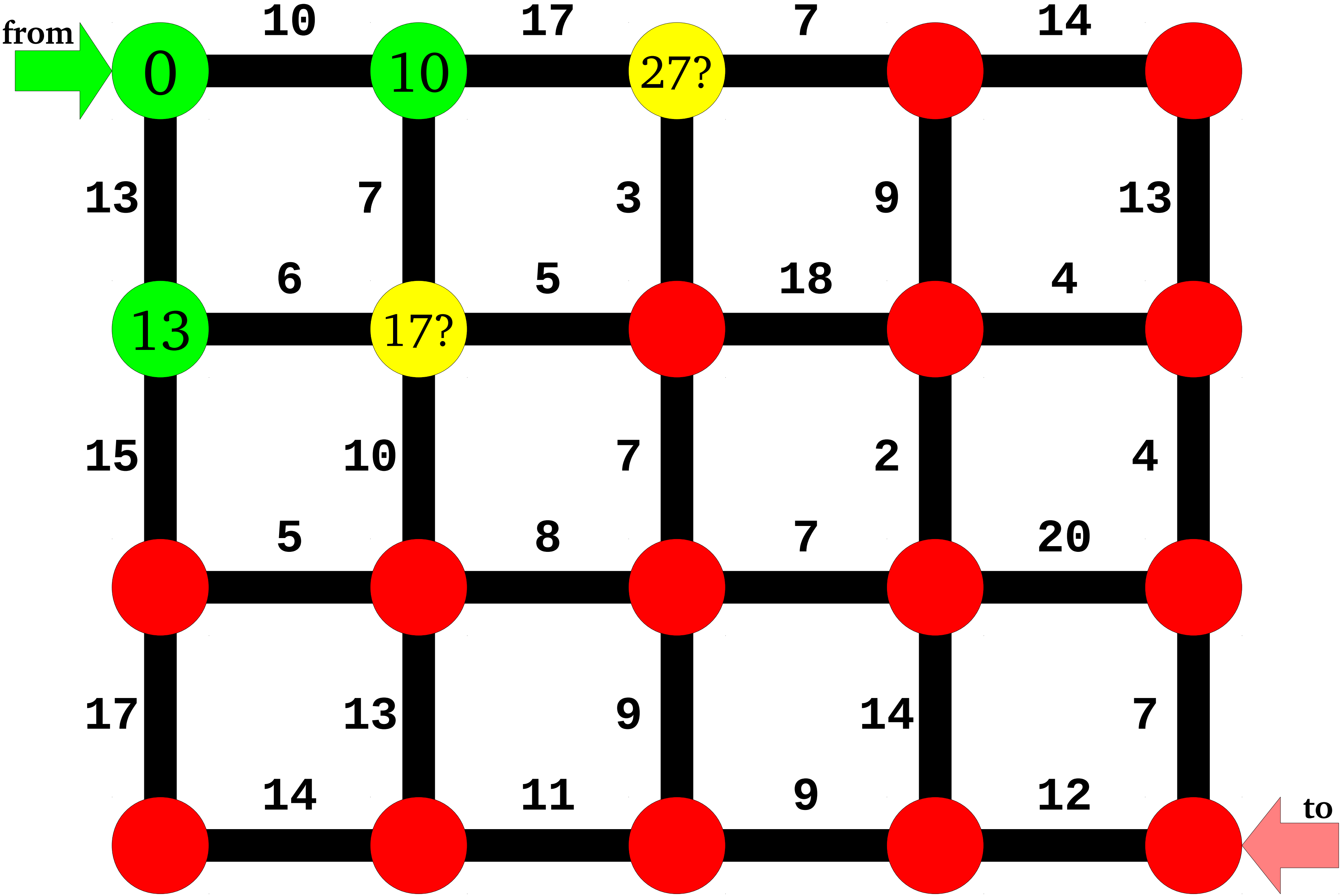

from
0
10
10?
17
7
14
13
7
3
9
13
6
5
18
4
13?
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

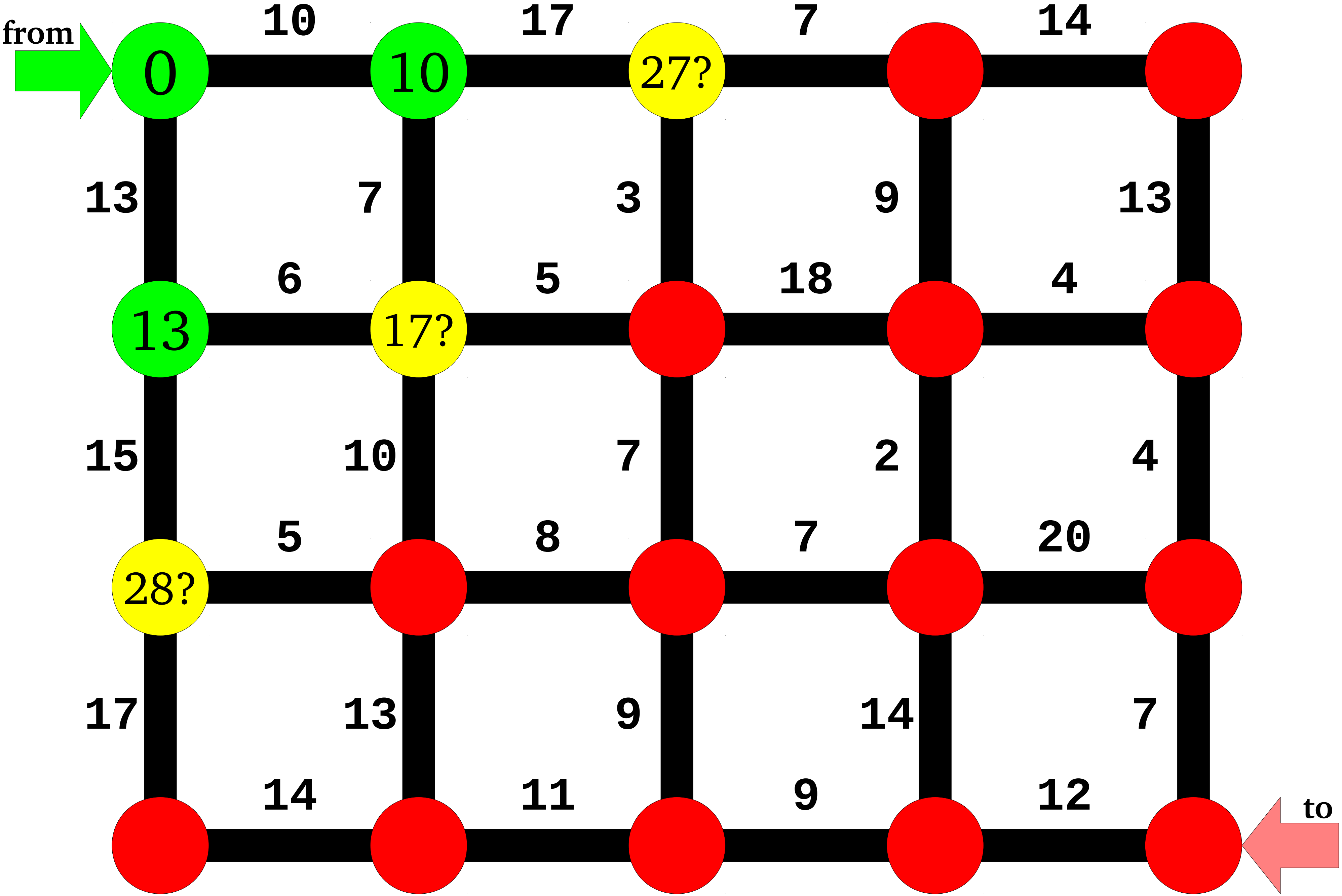

from
0
10
10?
17
7
14
13
7
3
9
13
6
13?
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

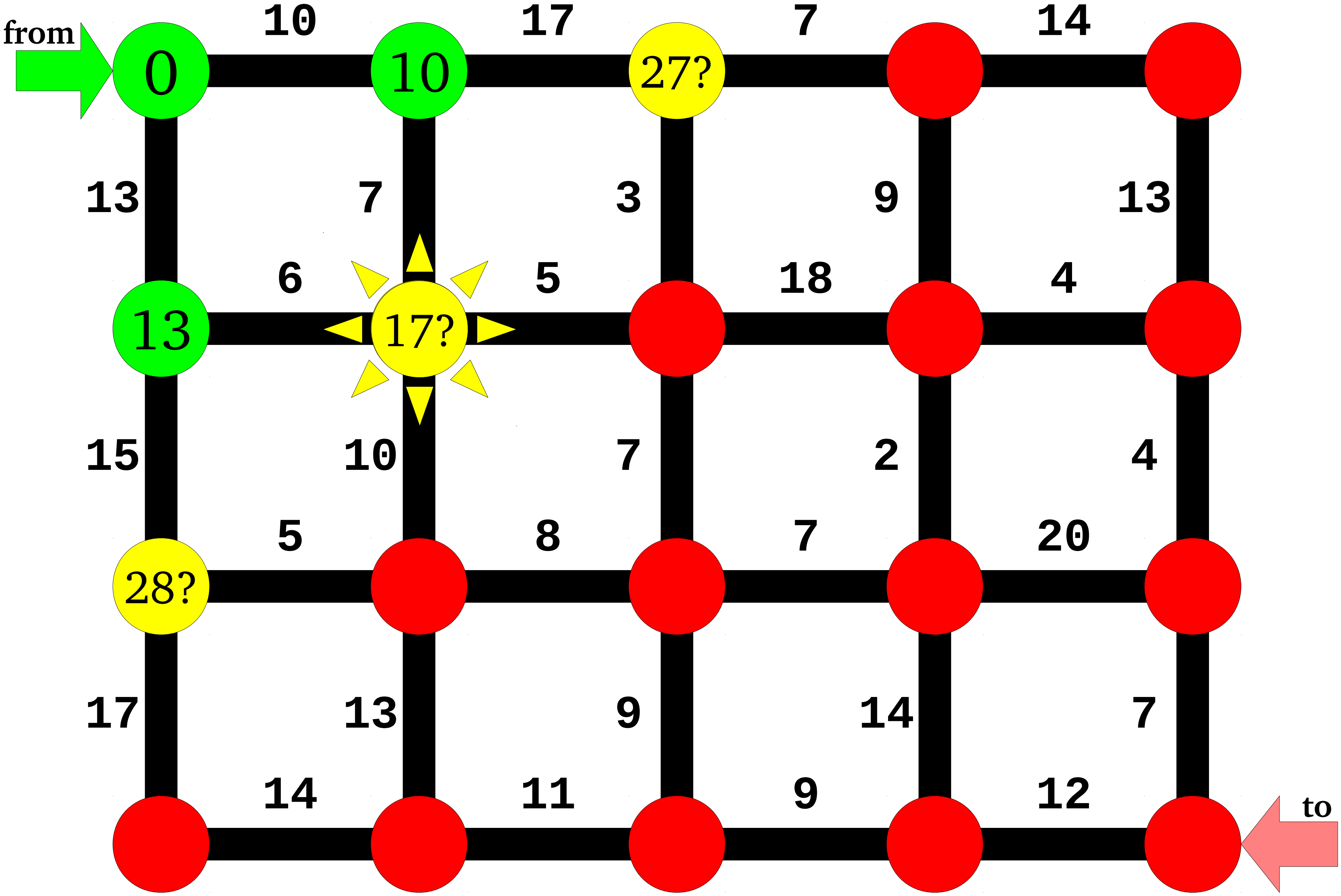

from
0
10
13?
10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

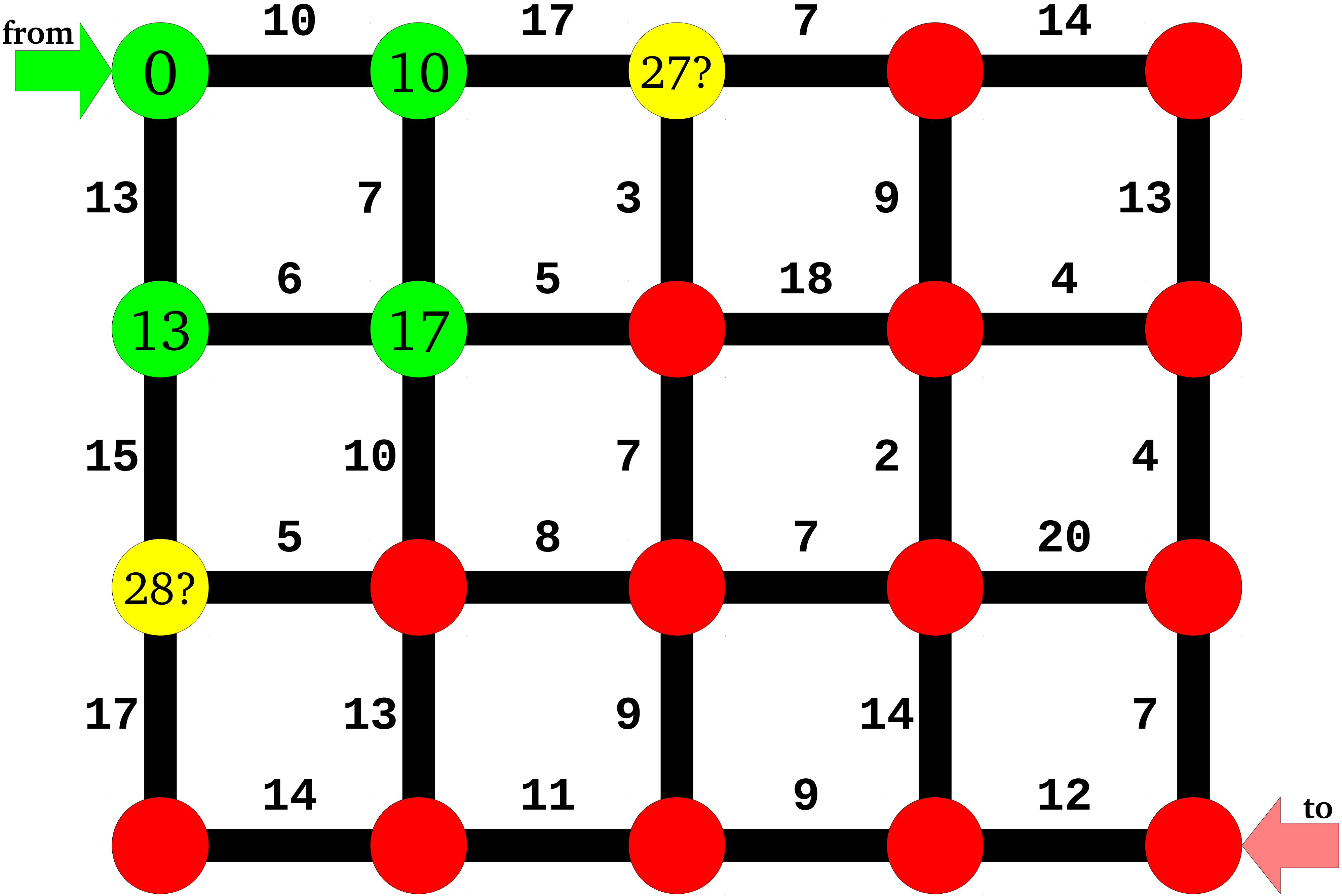

from
10
17
7
14
0
10
27?
13
7
3
9
13
6
5
18
4
13?
17?
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

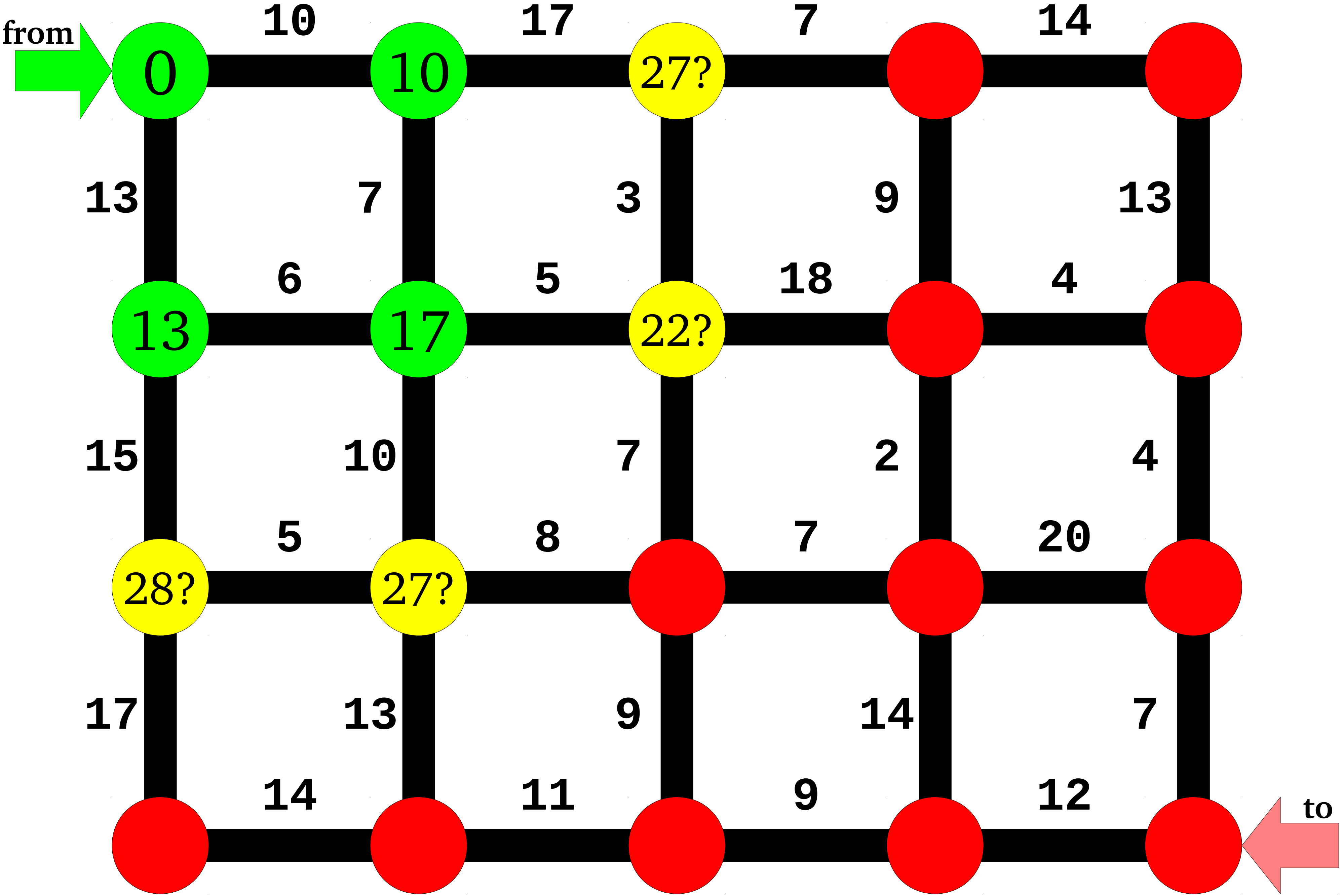

from
10
17
7
14
0
10
27?
13
7
3
9
13
6
5
18
4
13?
17?
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

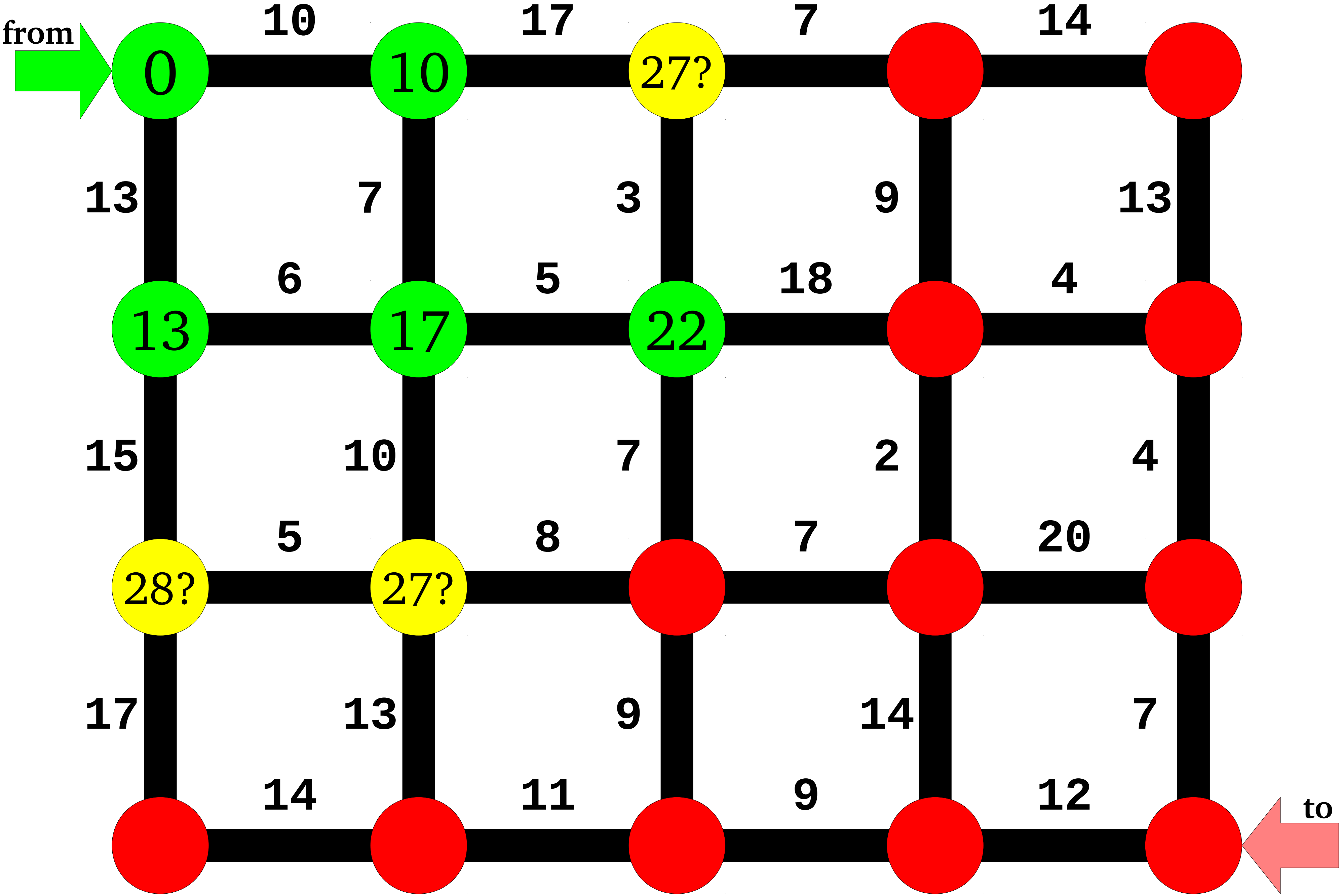

from
10
17
7
14
0
10
27?
13
7
3
9
13
6
5
18
4
13
17?
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12
to

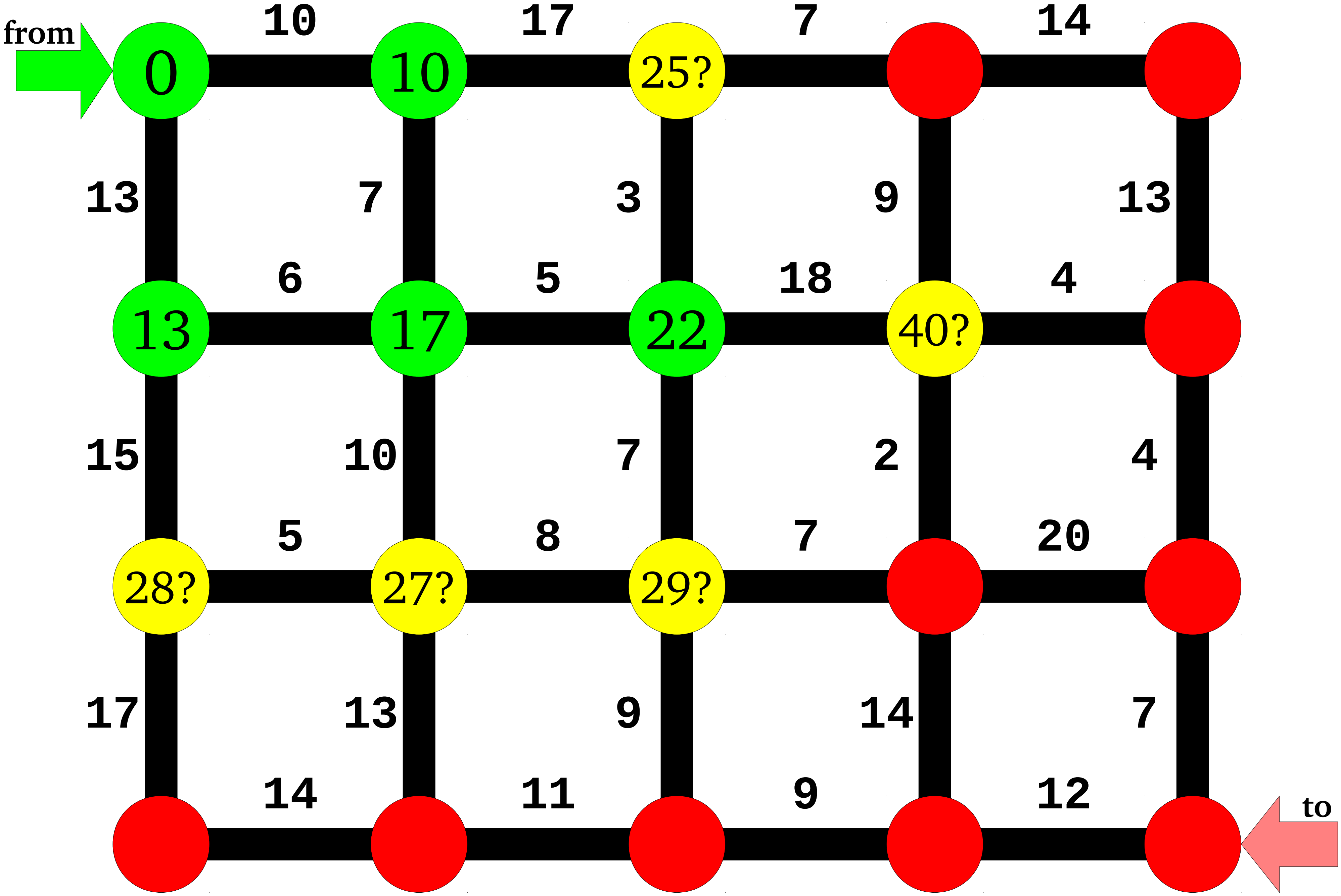

from
10
17
7
14
0
10
27?
13
7
3
9
13
6
5
18
4
13
17?
15
10
7
2
4
5
8
7
20
28?
17
13
9
14
7
14
11
9
12
to

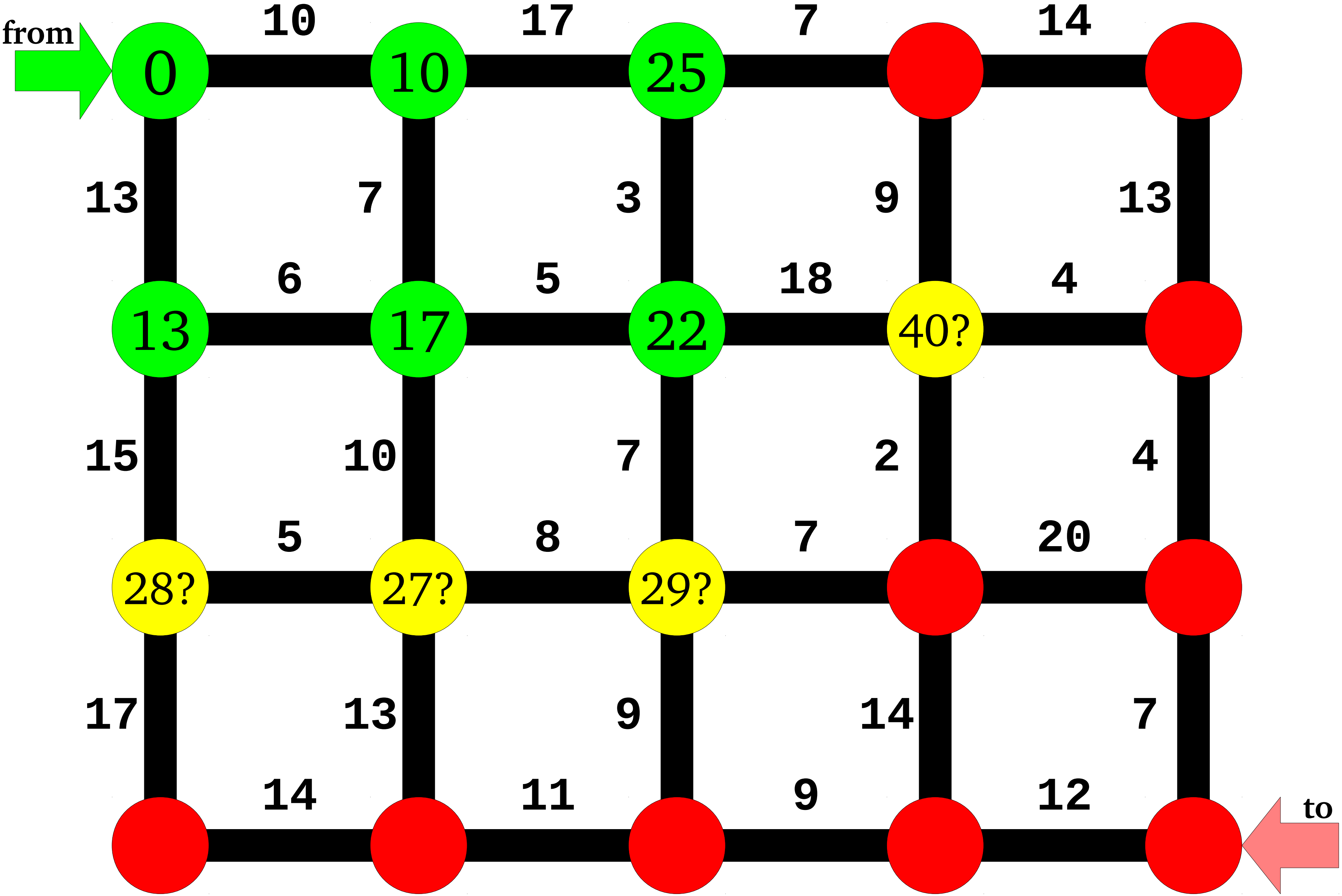

from
10
17
7
14
0
10
27?
13
7
3
9
13
6
5
18
4
13
17?
15
10
7
2
4
5
8
7
20
28?
17
13
9
14
7
14
11
9
12
to

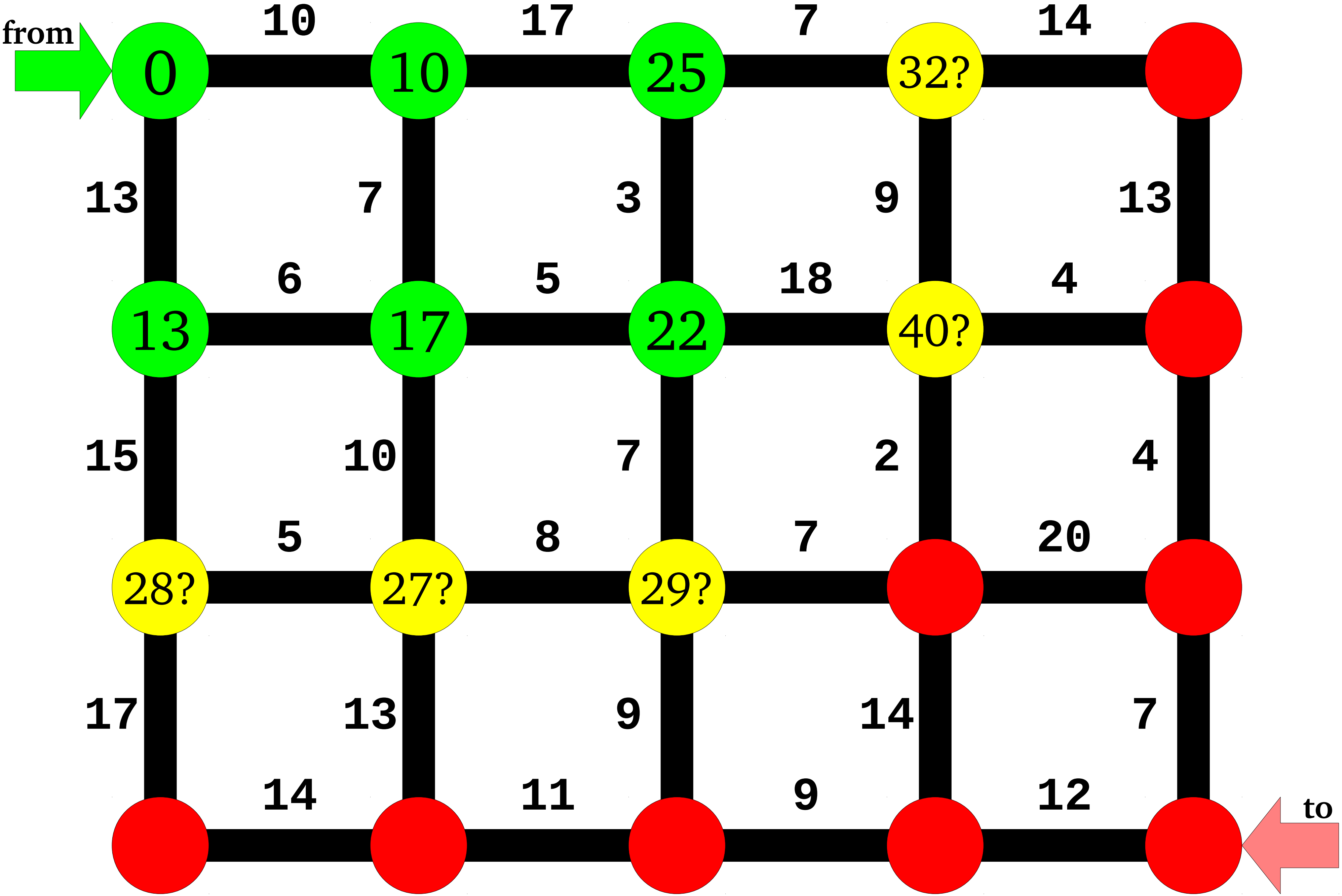

from
10
17
7
14
0
10
27?
13
7
3
9
13
6
5
18
4
13
17
15
10
7
2
4
5
8
7
20
28?
17
13
9
14
7
14
11
9
12
to

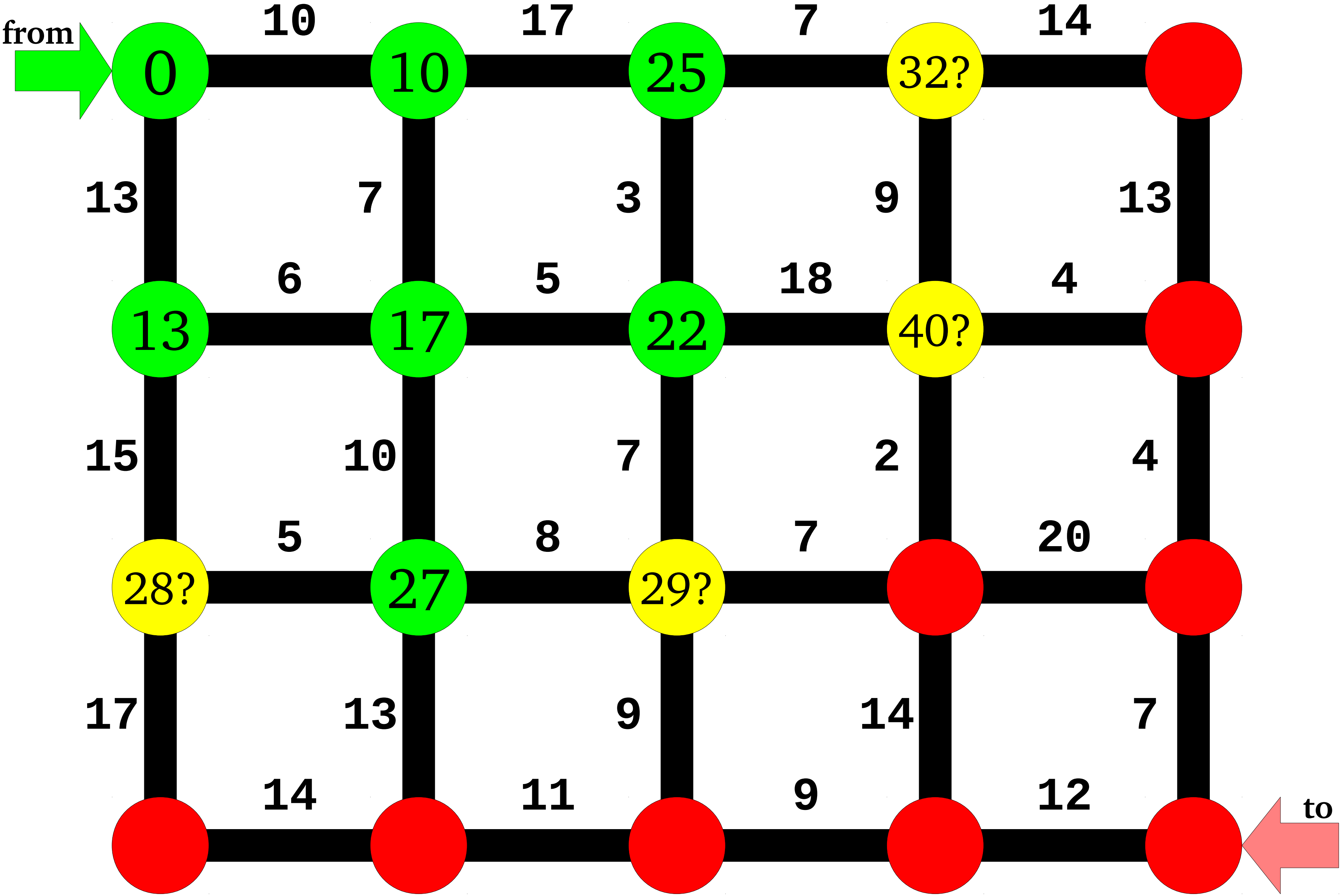

from
10
17
7
14
0
10
27?
13
7
3
9
13
6
5
18
4
13
17
22?
15
10
7
2
4
5
8
7
20
28?
27?
17
13
9
14
7
14
11
9
12
to

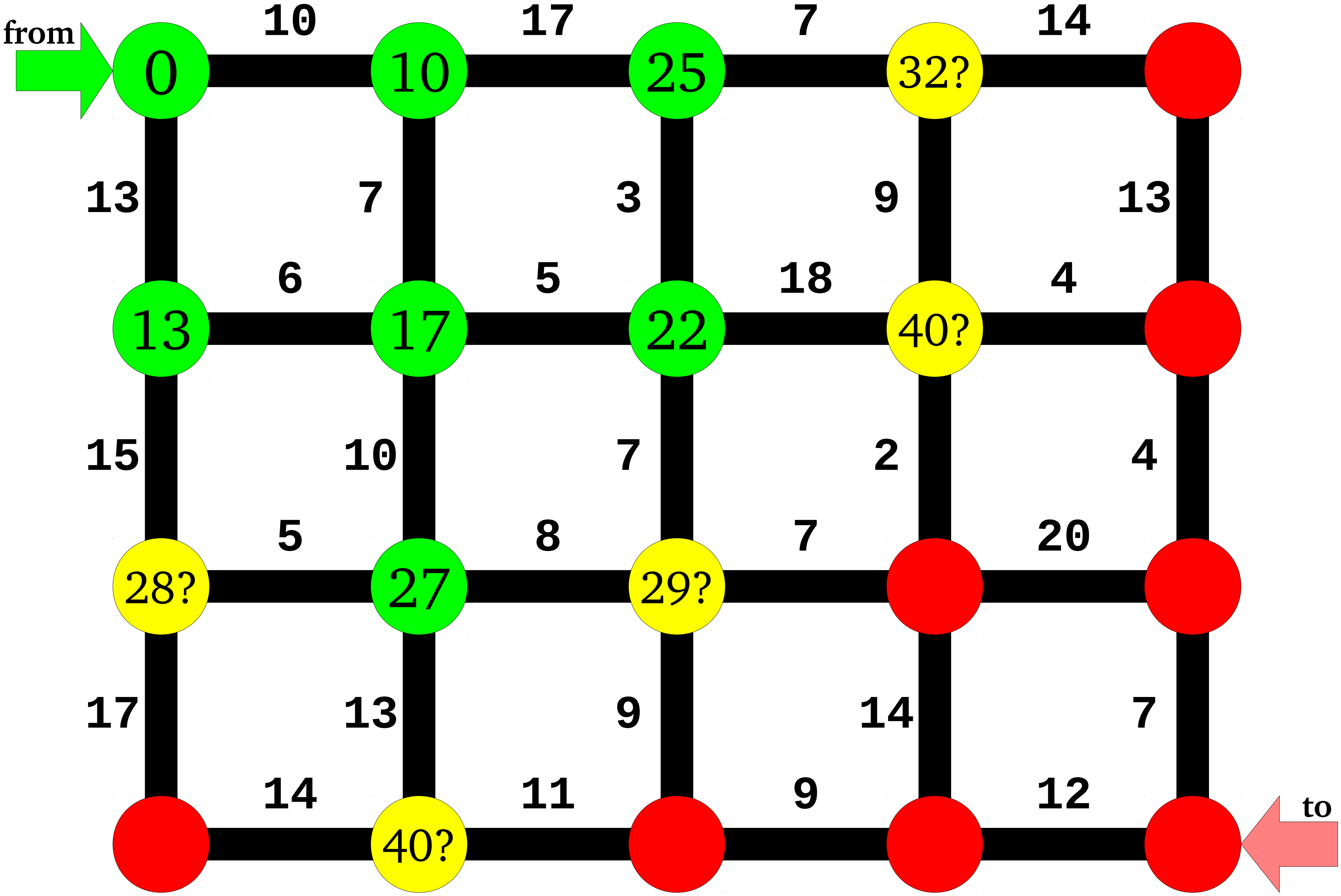

from
10
17
7
14
0
10
27?
13
7
3
9
13
6
5
18
4
13
17
22
15
10
7
2
4
5
8
7
20
28?
27?
17
13
9
14
7
14
11
9
12
to

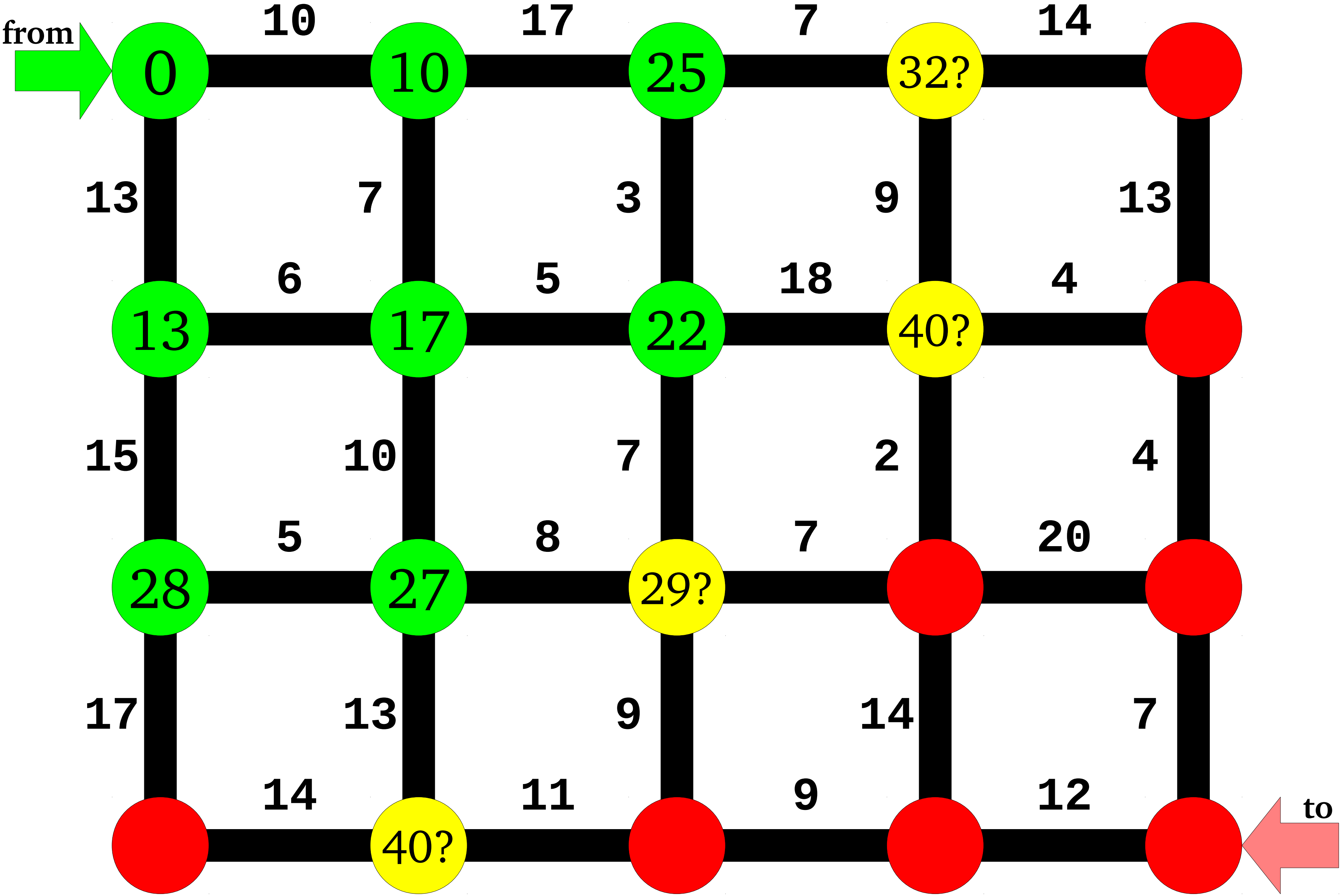

from
10
17
7
14
0
10
25?
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28?
27?
29?
17
13
9
14
7
14
11
9
12
to

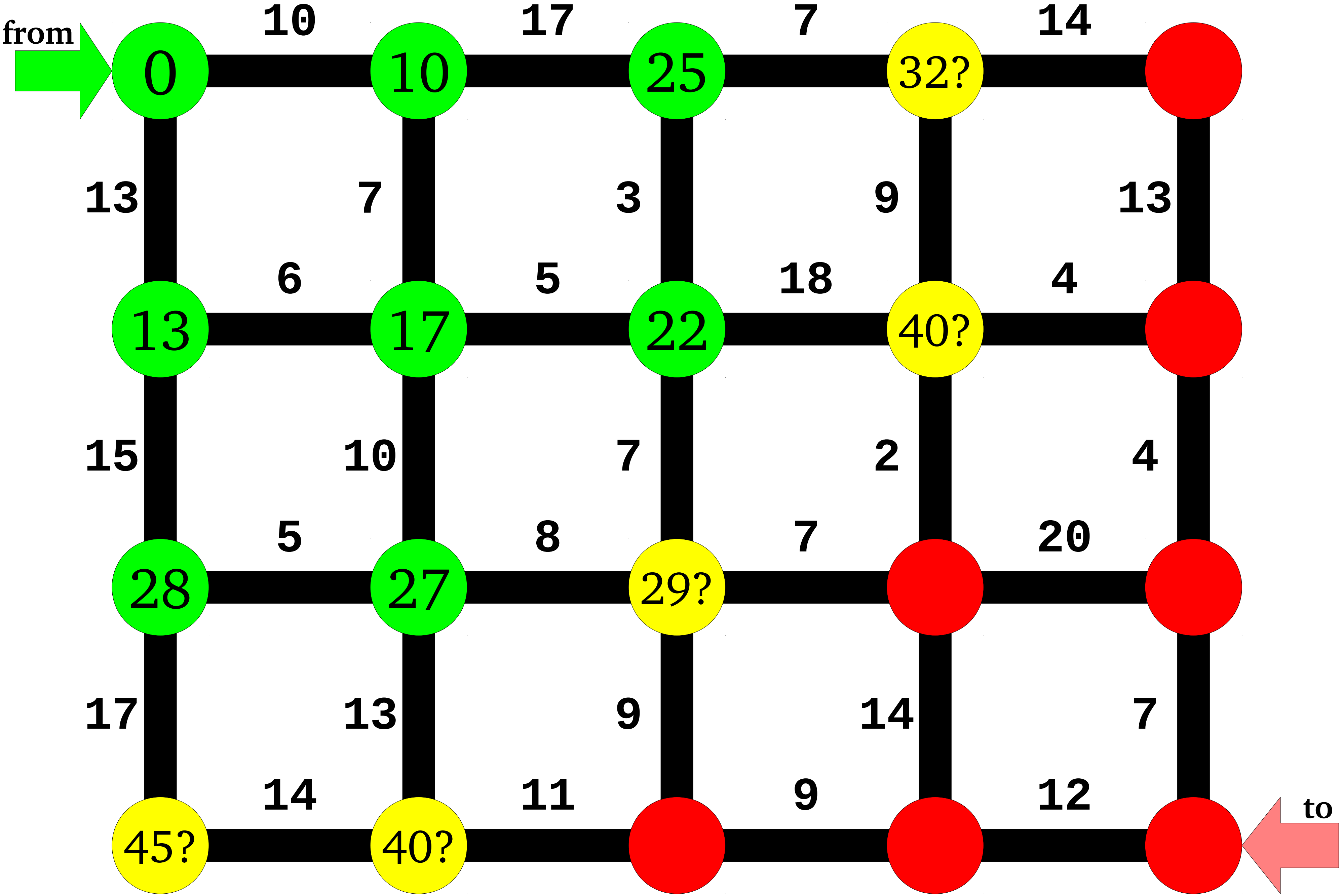

from
10
17
7
14
0
10
25
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28?
27?
29?
17
13
9
14
7
14
11
9
12
to

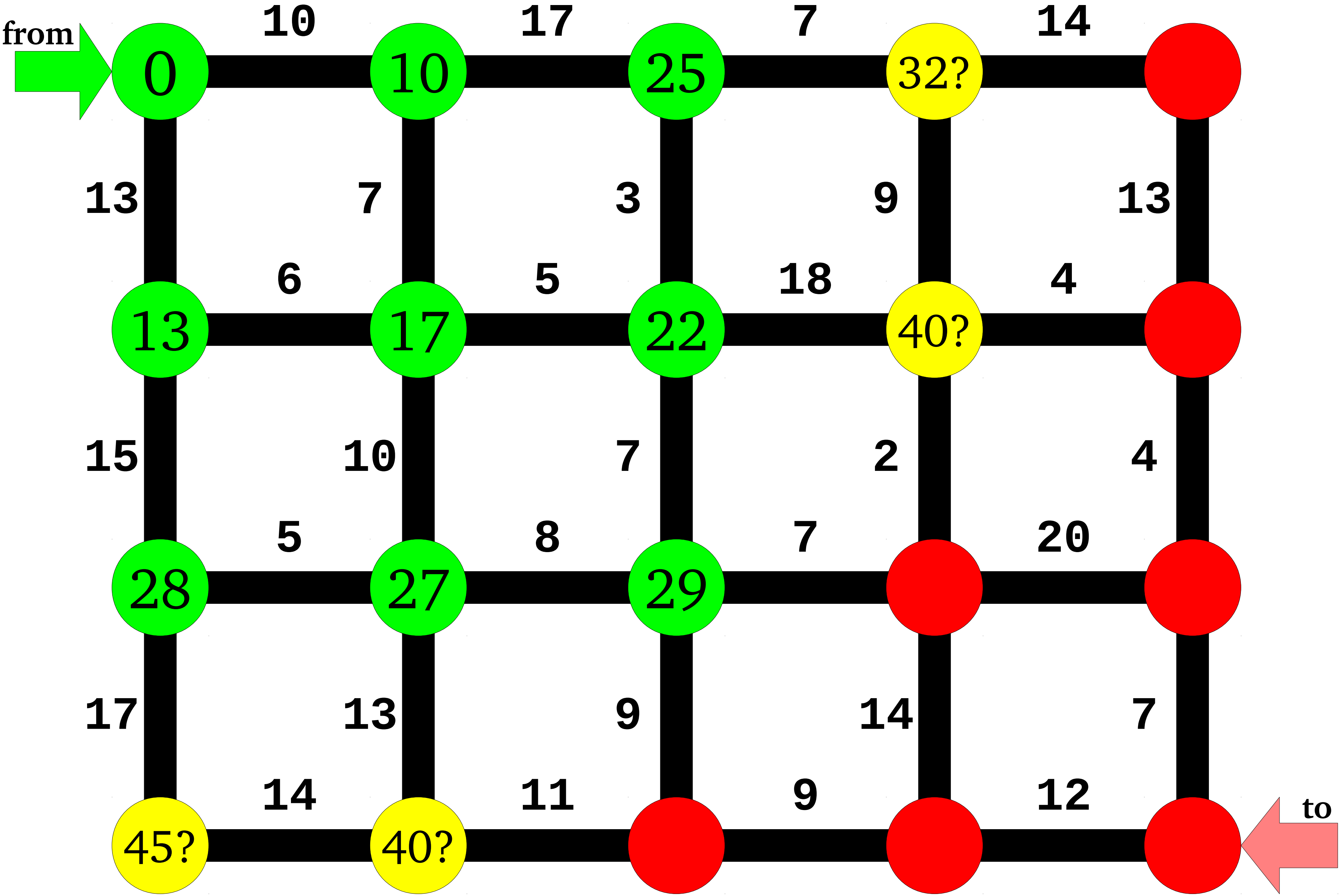

from
10
17
7
14
0
10
25
32?
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28?
27?
29?
17
13
9
14
7
14
11
9
12
to

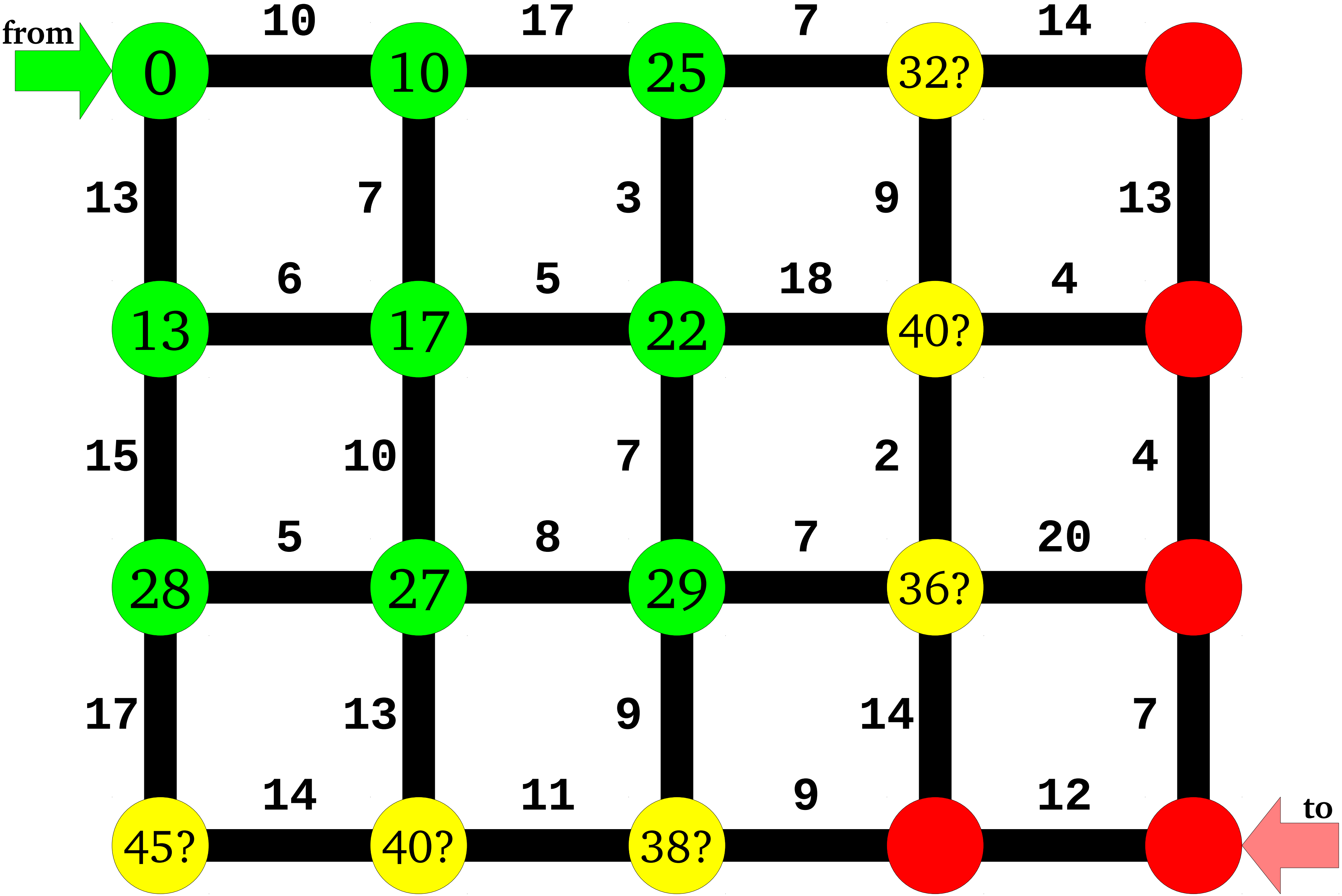

from
10
17
7
14
0
10
25
32?
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28?
27
29?
17
13
9
14
7
14
11
9
12
to

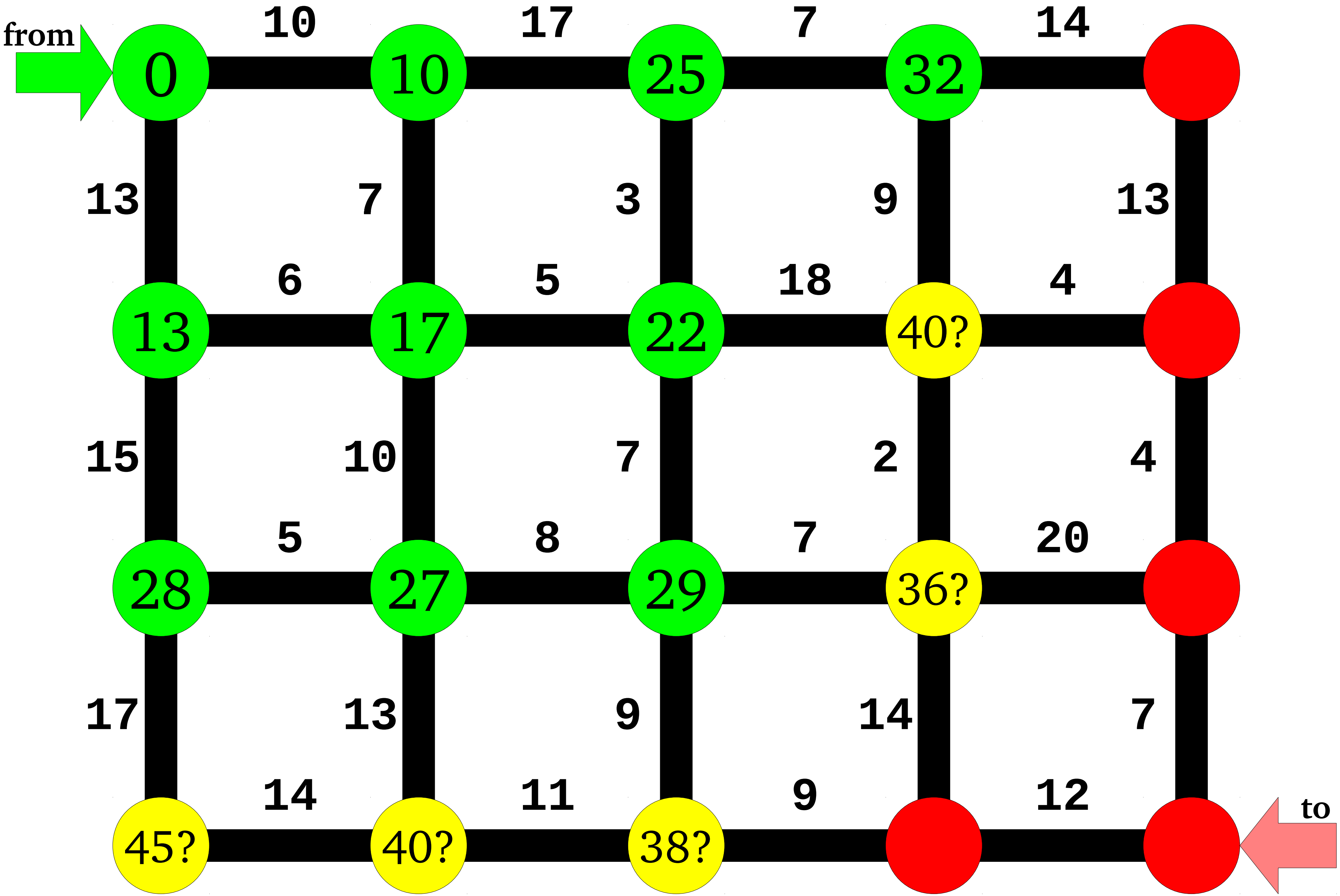

from
10
17
7
14
0
10
25
32?
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28?
27
29?
17
13
9
14
7
14
11
9
12
to
40?

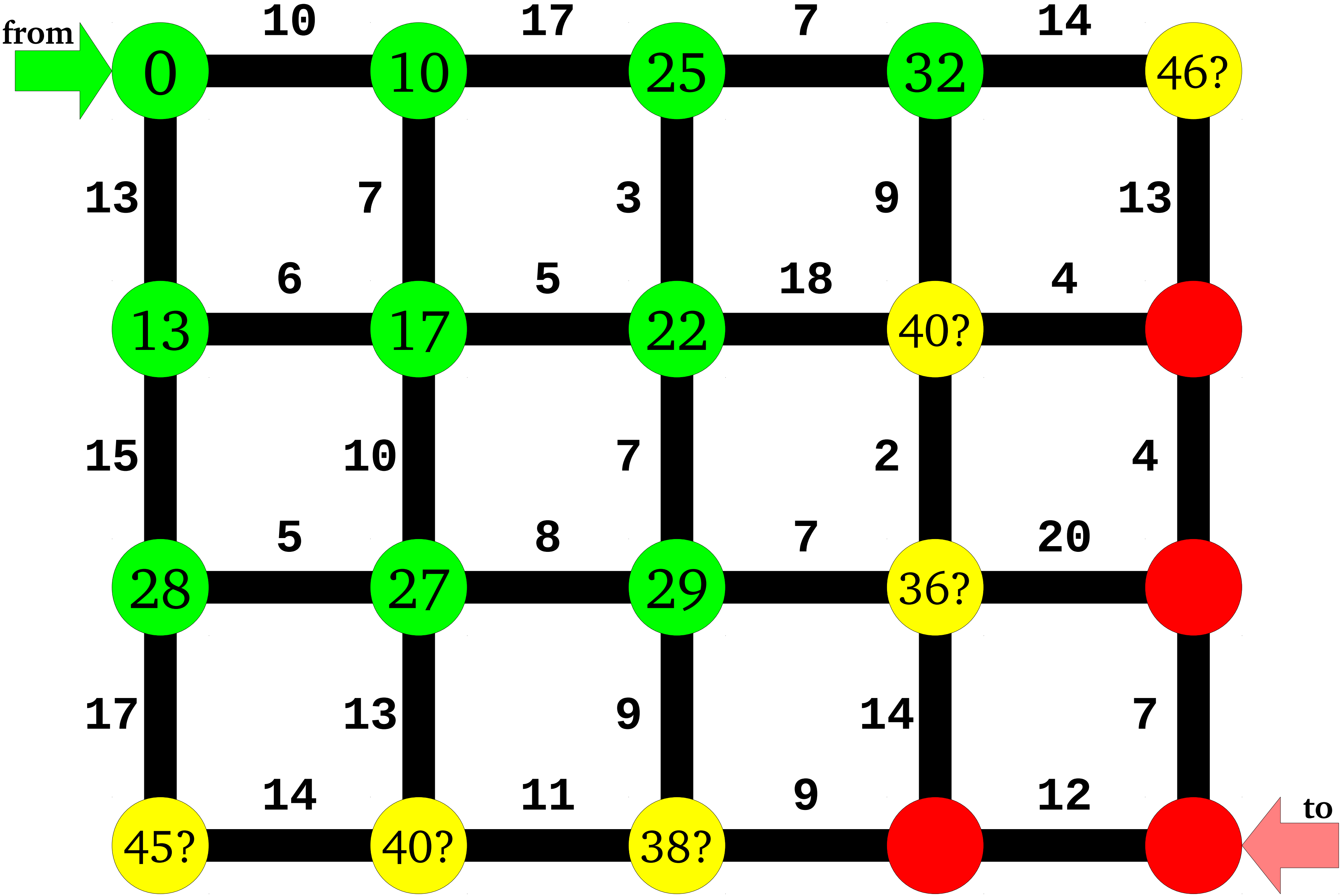

from
10
17
7
14
0
10
25
32?
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28
27
29?
17
13
9
14
7
14
11
9
12
to
40?

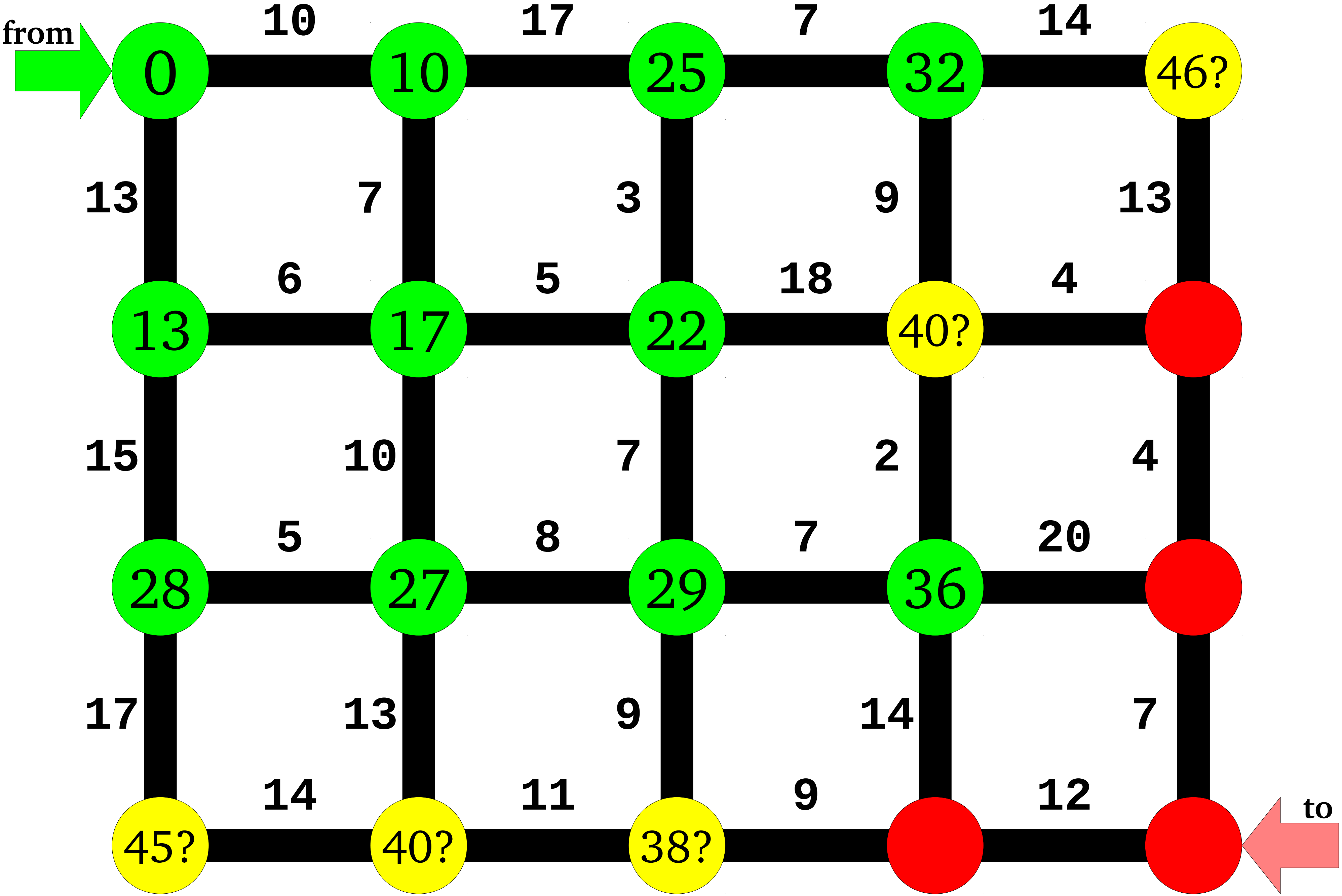

from
10
17
7
14
0
10
25
32?
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28
27
29?
17
13
9
14
7
14
11
9
12
to
45?
40?

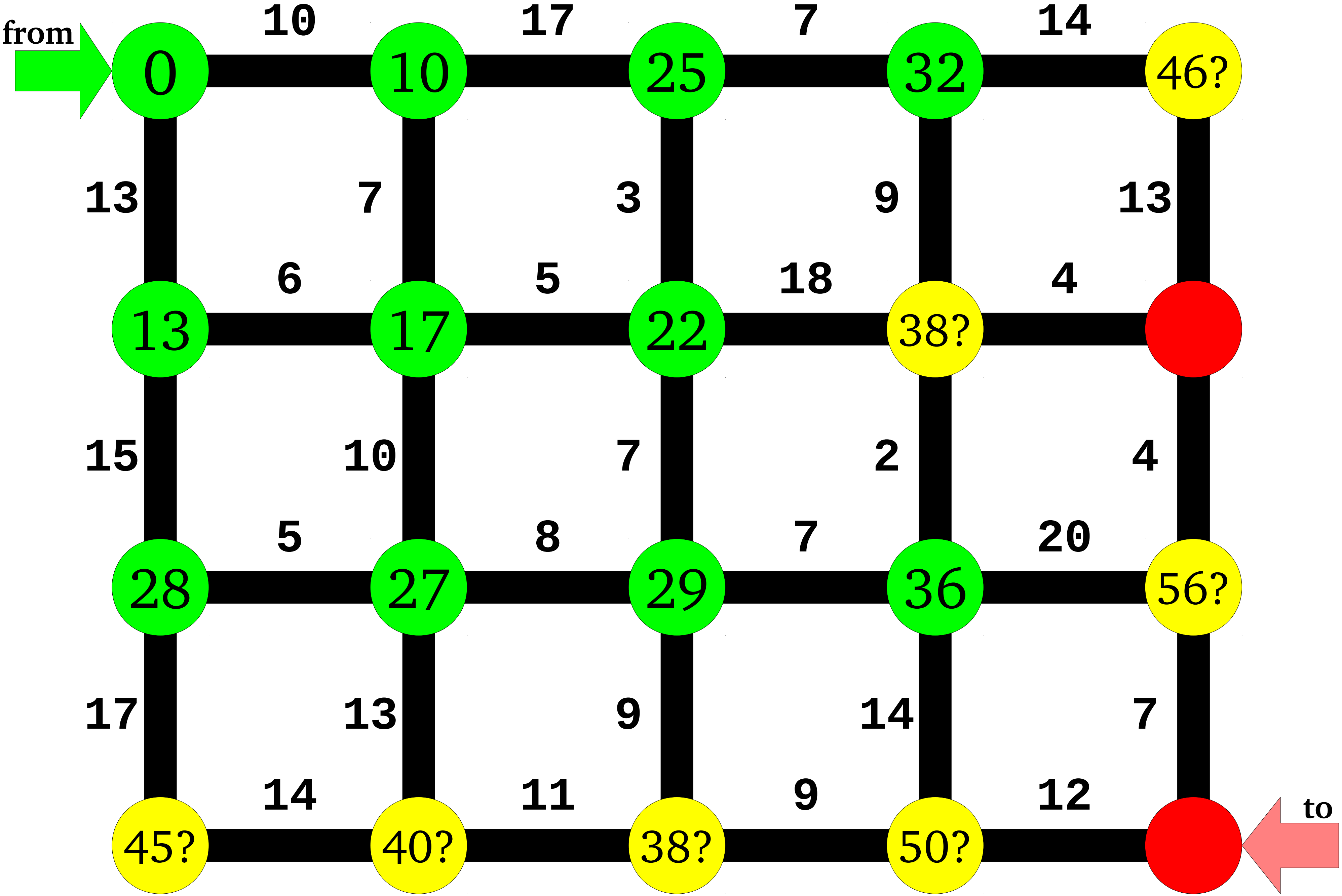

from
10
17
7
14
0
10
25
32?
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28
27
29
17
13
9
14
7
14
11
9
12
to
45?
40?

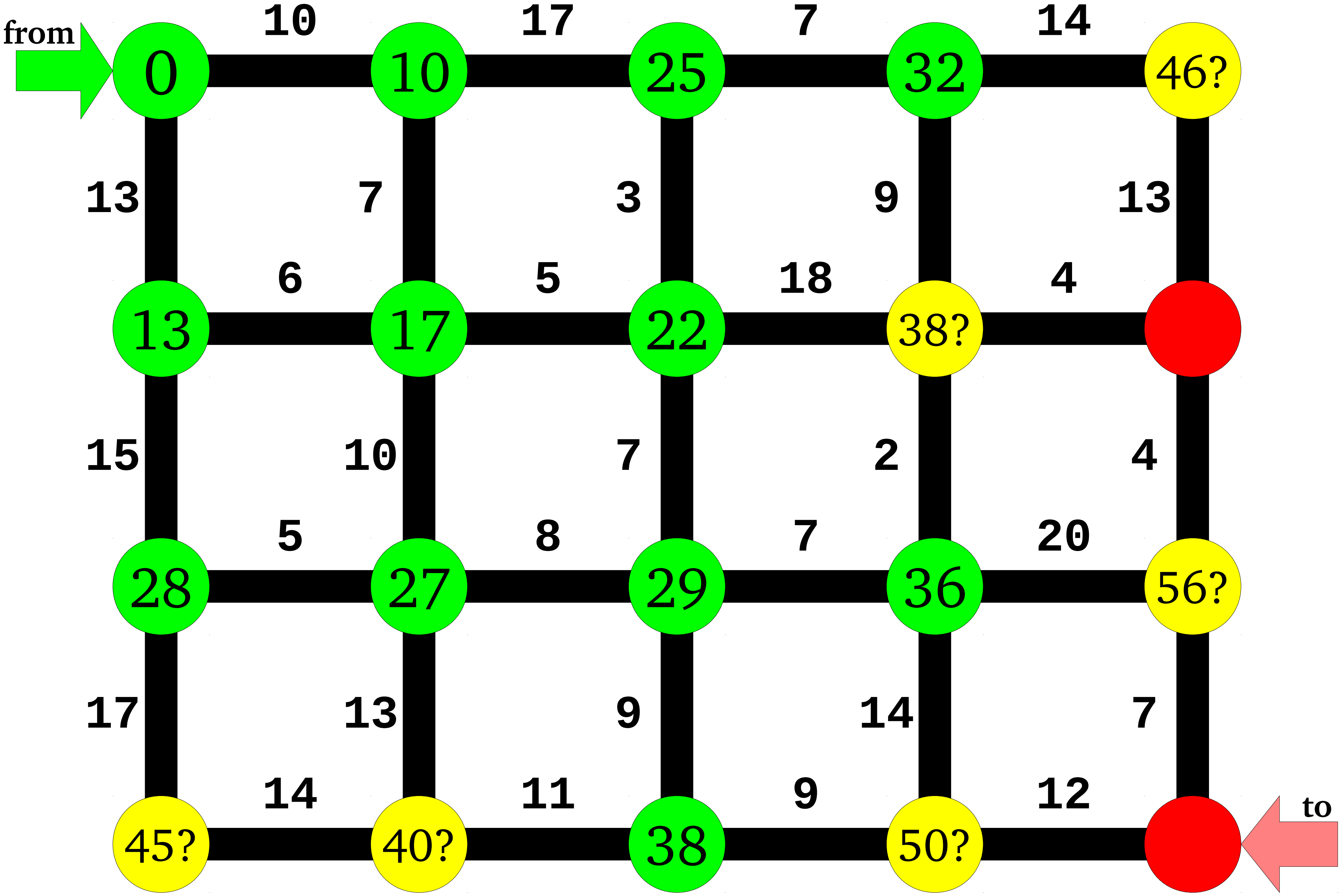

from
10
17
7
14
0
10
25
32?
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28
27
29
36?
17
13
9
14
7
14
11
9
12
to
45?
40?
38?

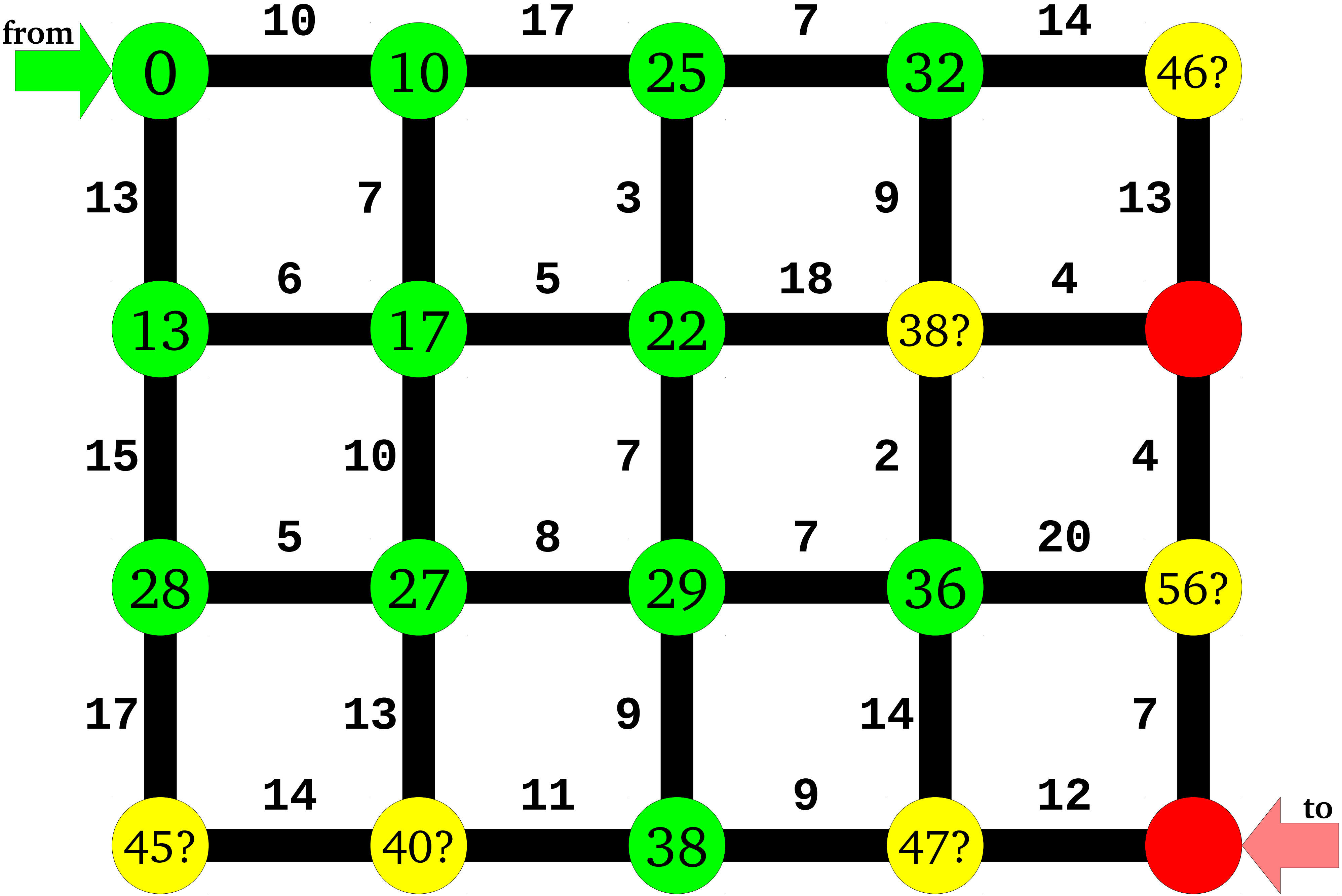

from
10
17
7
14
0
10
25
32
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28
27
29
36?
17
13
9
14
7
14
11
9
12
to
45?
40?
38?

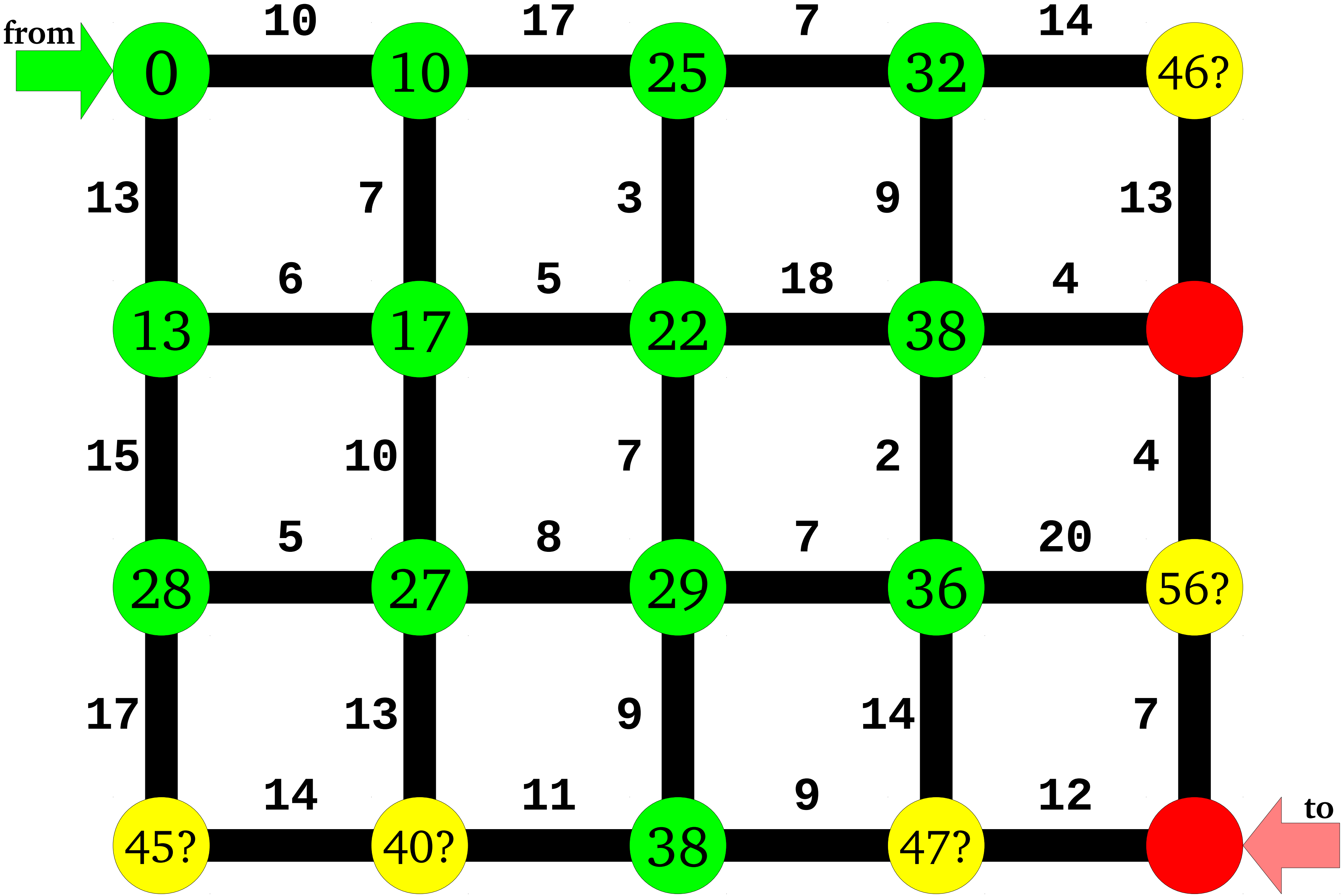

from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28
27
29
36?
17
13
9
14
7
14
11
9
12
45?
40?
38?
to

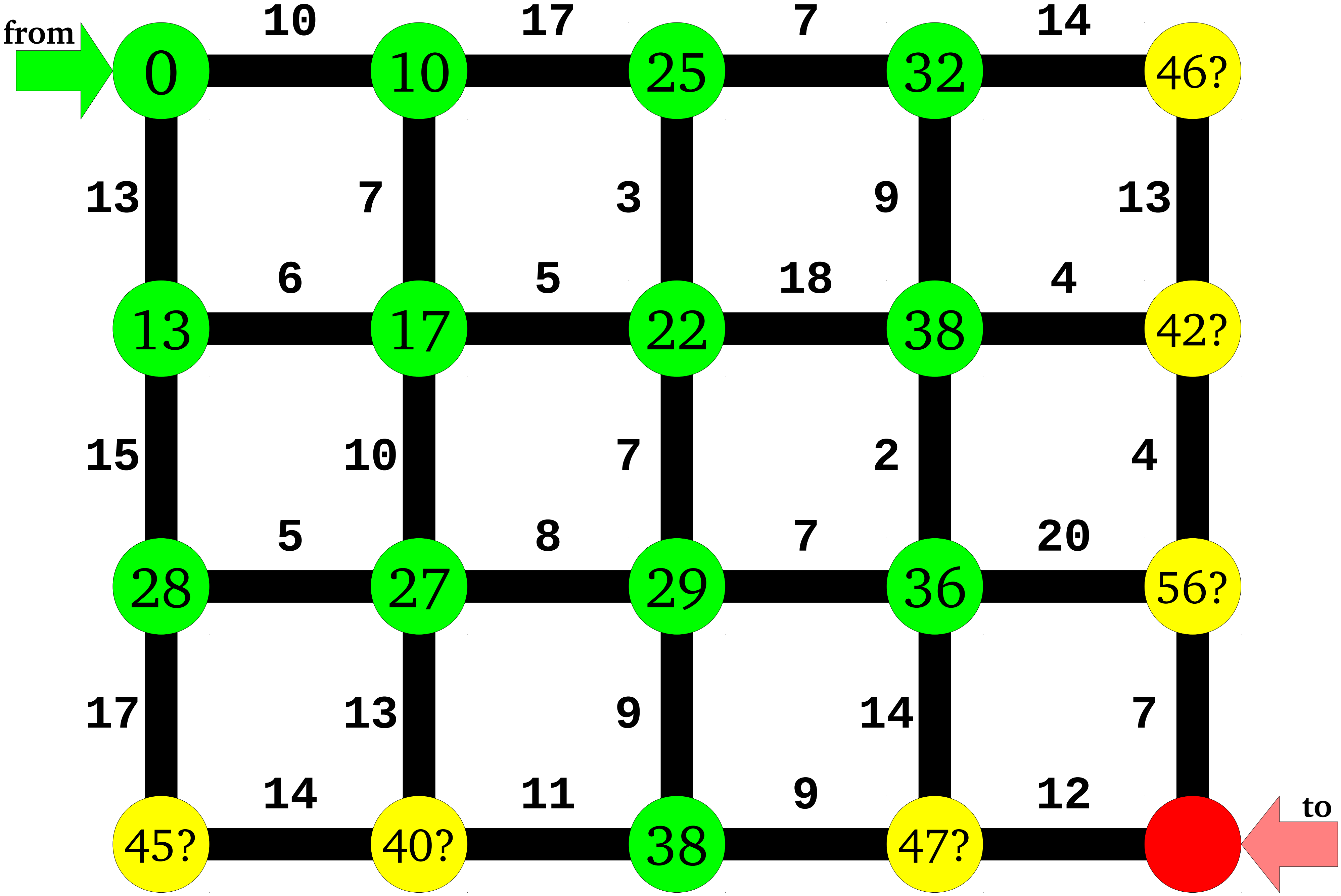

from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
40?
15
10
7
2
4
5
8
7
20
28
27
29
36
17
13
9
14
7
14
11
9
12
to
45?
40?
38?

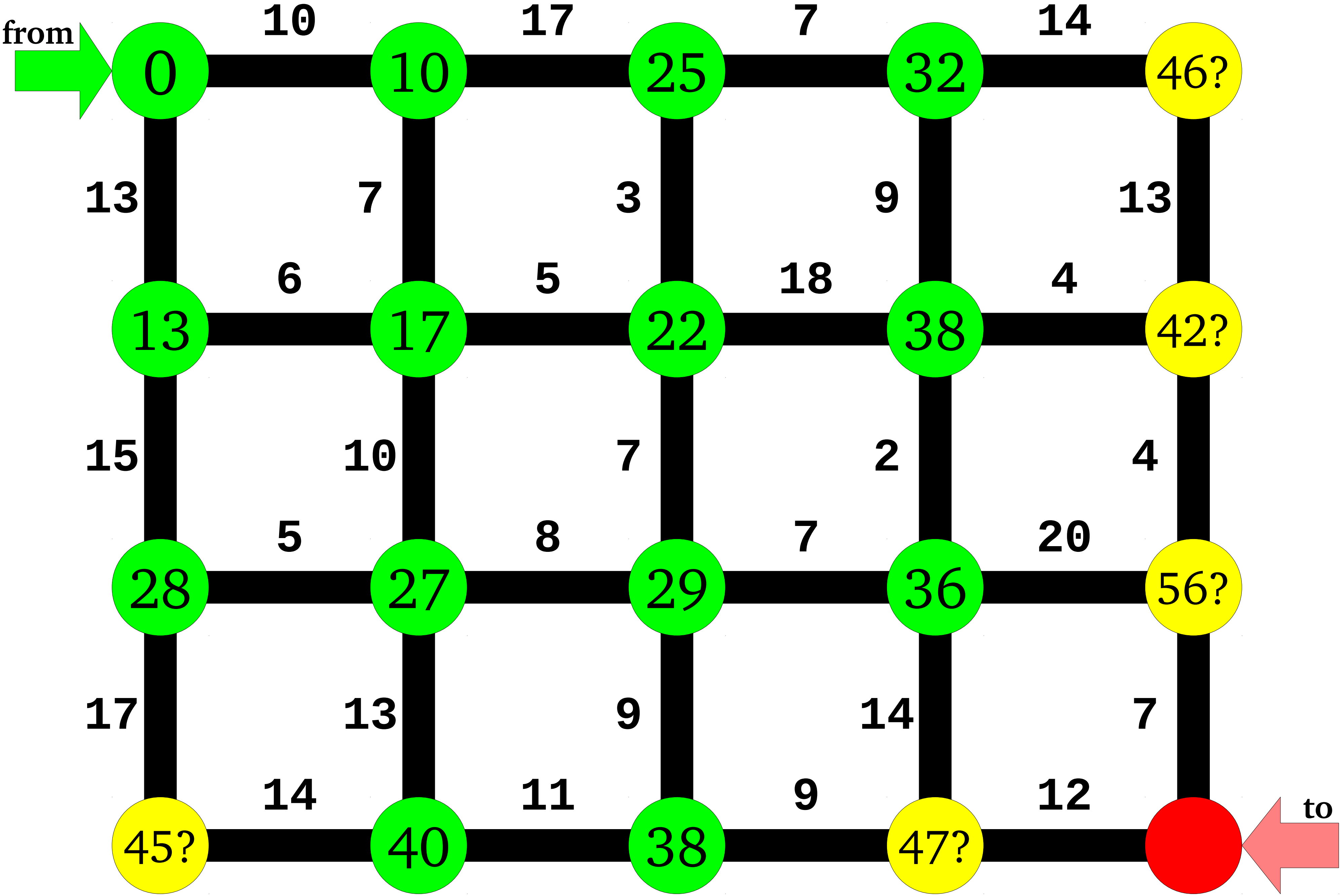

from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
38?
15
10
7
2
4
5
8
7
20
28
27
29
36
56?
17
13
9
14
7
14
11
9
12
to
45?
40?
38?
50?

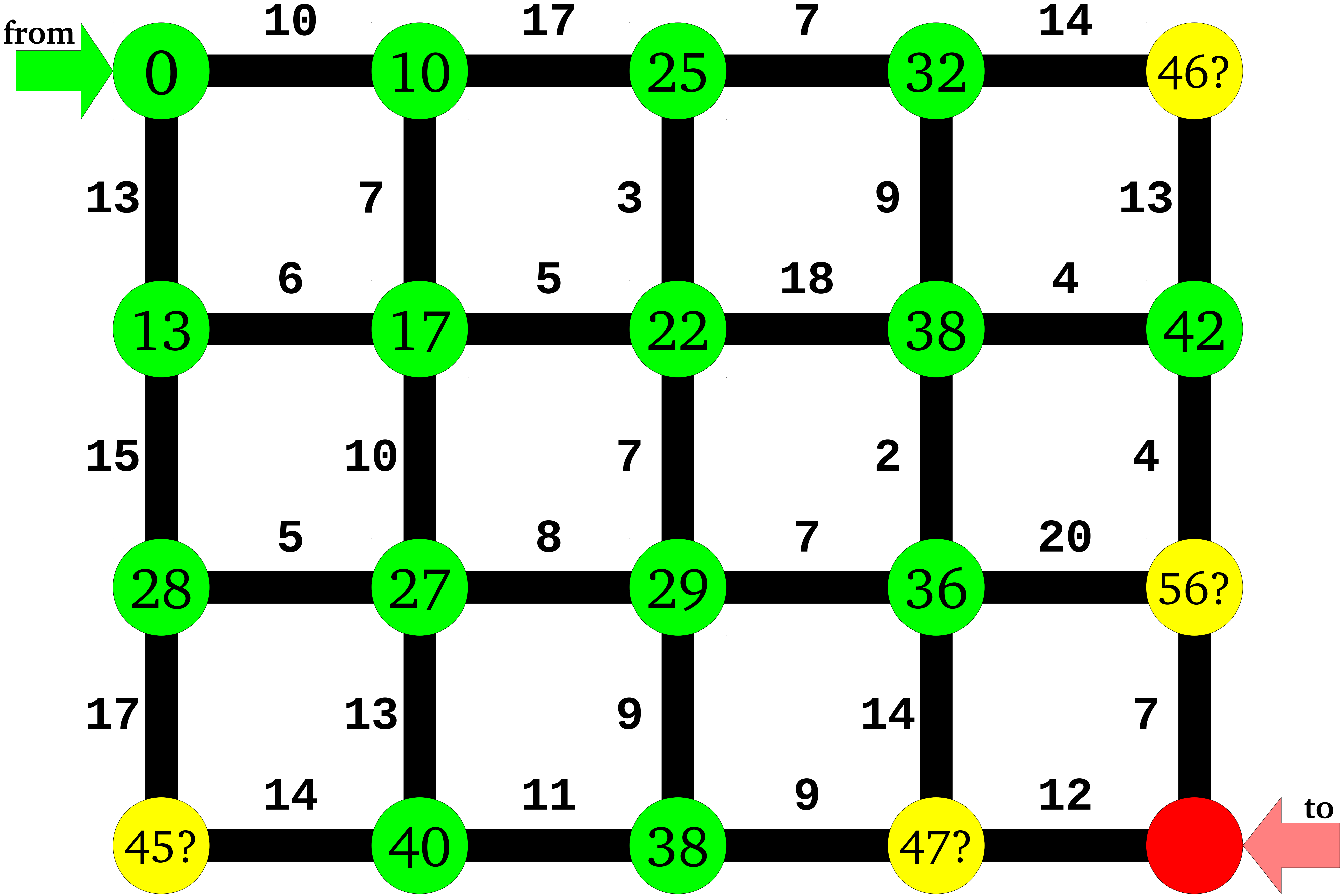

from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
38?
15
10
7
2
4
5
8
7
20
28
27
29
36
56?
17
13
9
14
7
14
11
9
12
to
45?
40?
38
50?

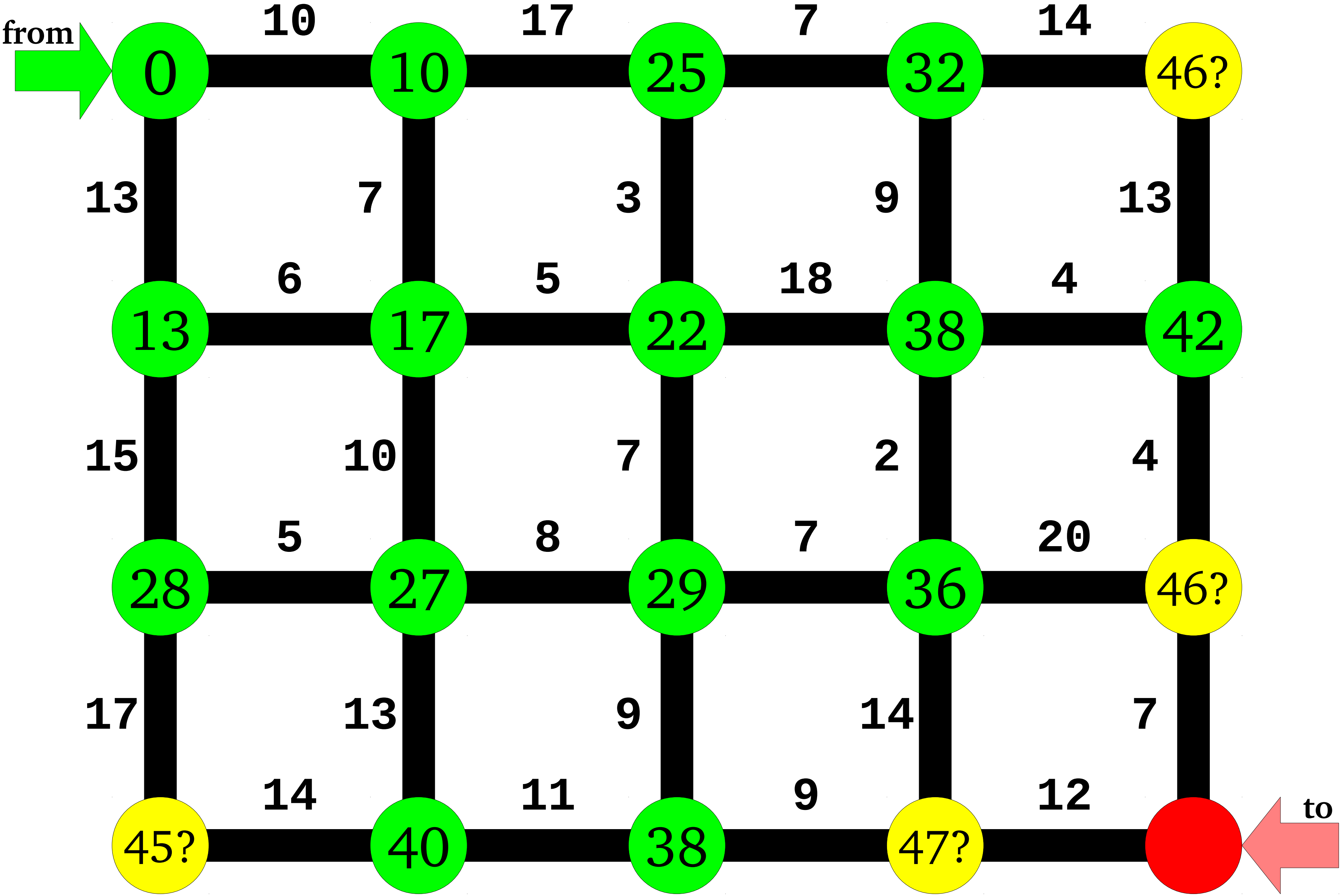

from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
38?
15
10
7
2
4
5
8
7
20
28
27
29
36
56?
17
13
9
14
7
14
11
9
12
to
45?
40?
38
47?

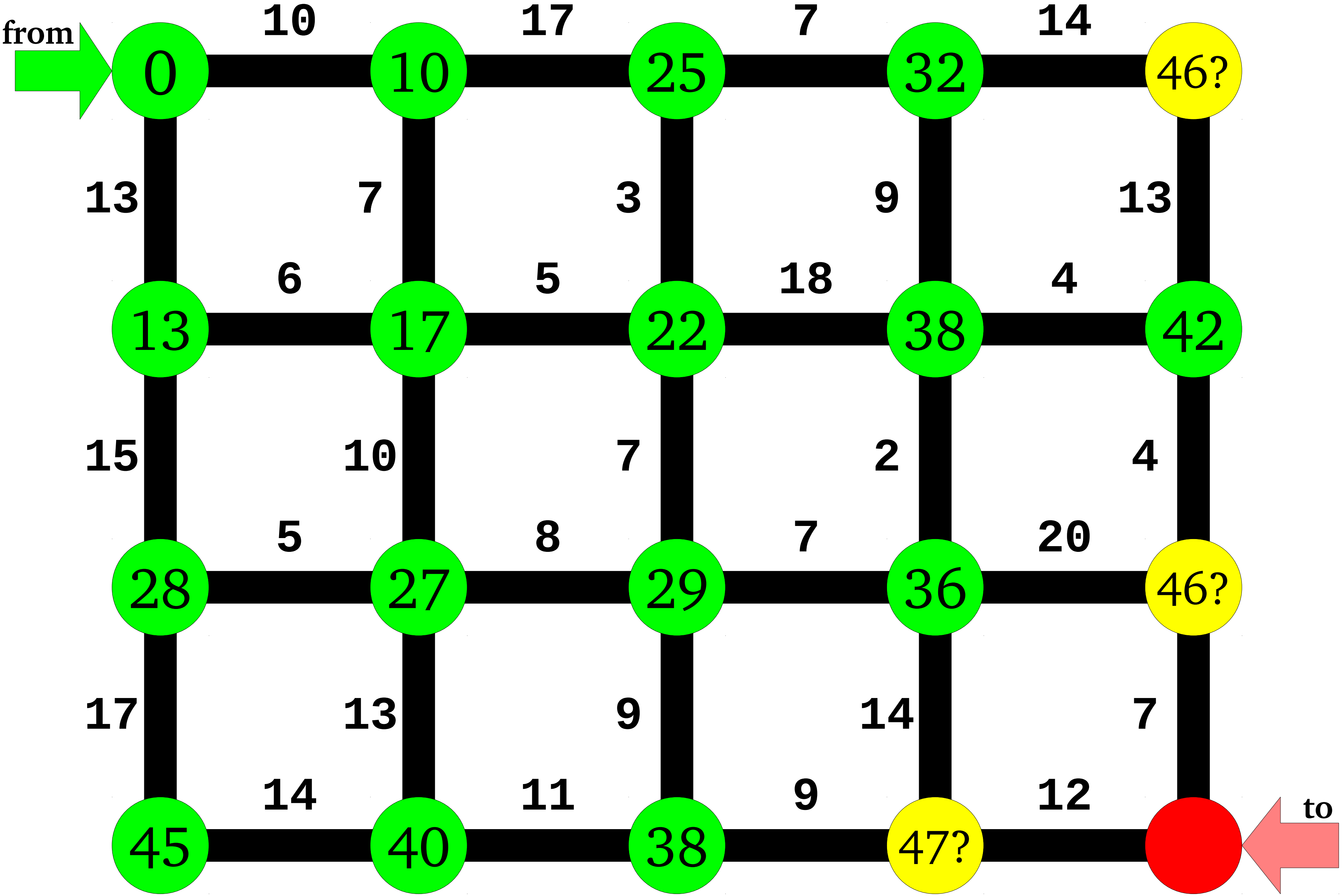

from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
38
15
10
7
2
4
5
8
7
20
28
27
29
36
56?
17
13
9
14
7
14
11
9
12
to
45?
40?
38
47?

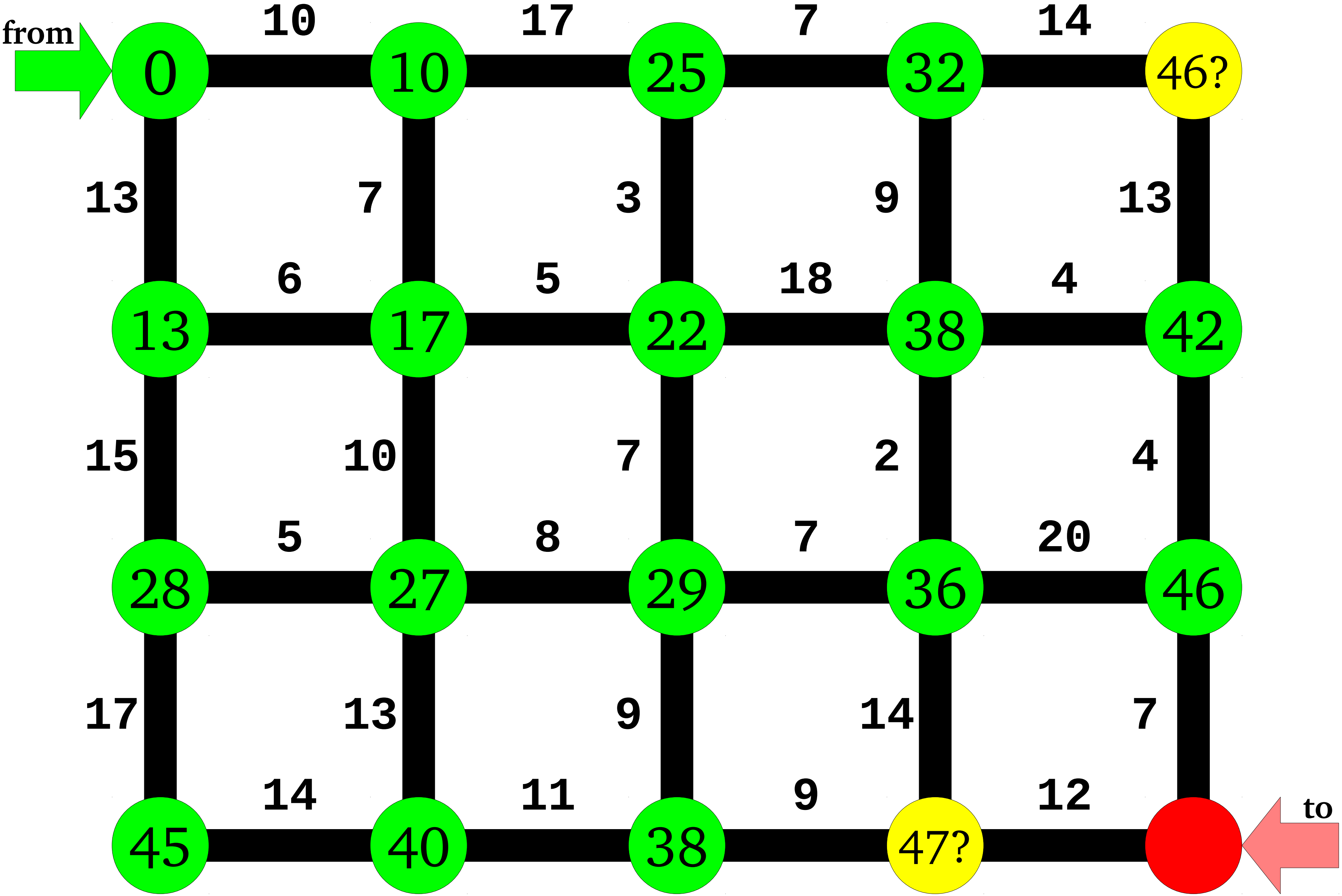

from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
38
42?
15
10
7
2
4
5
8
7
20
28
27
29
36
56?
17
13
9
14
7
14
11
9
12
to
45?
40?
38
47?

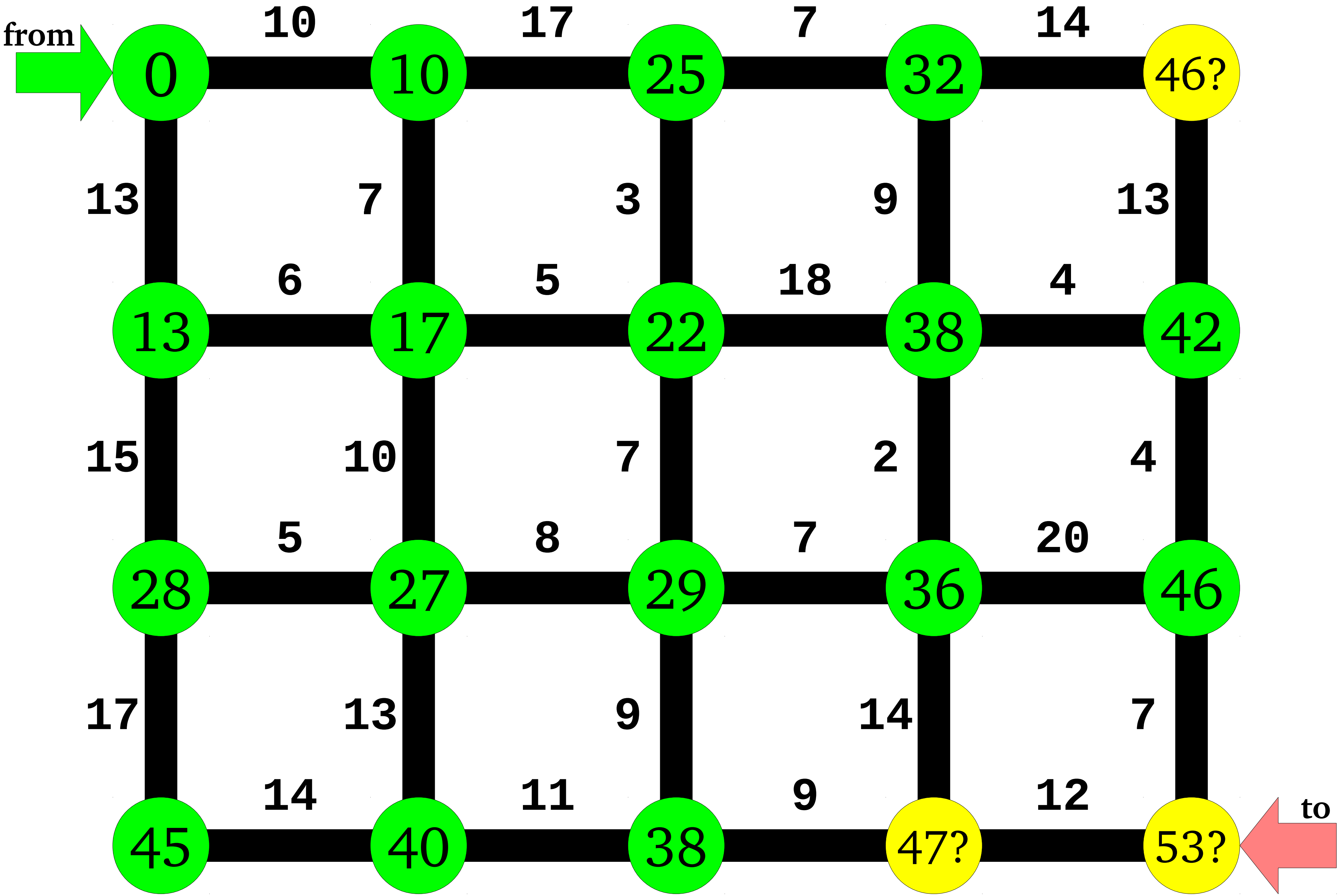

from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
38
42?
15
10
7
2
4
5
8
7
20
28
27
29
36
56?
17
13
9
14
7
14
11
9
12
to
45?
40
38
47?

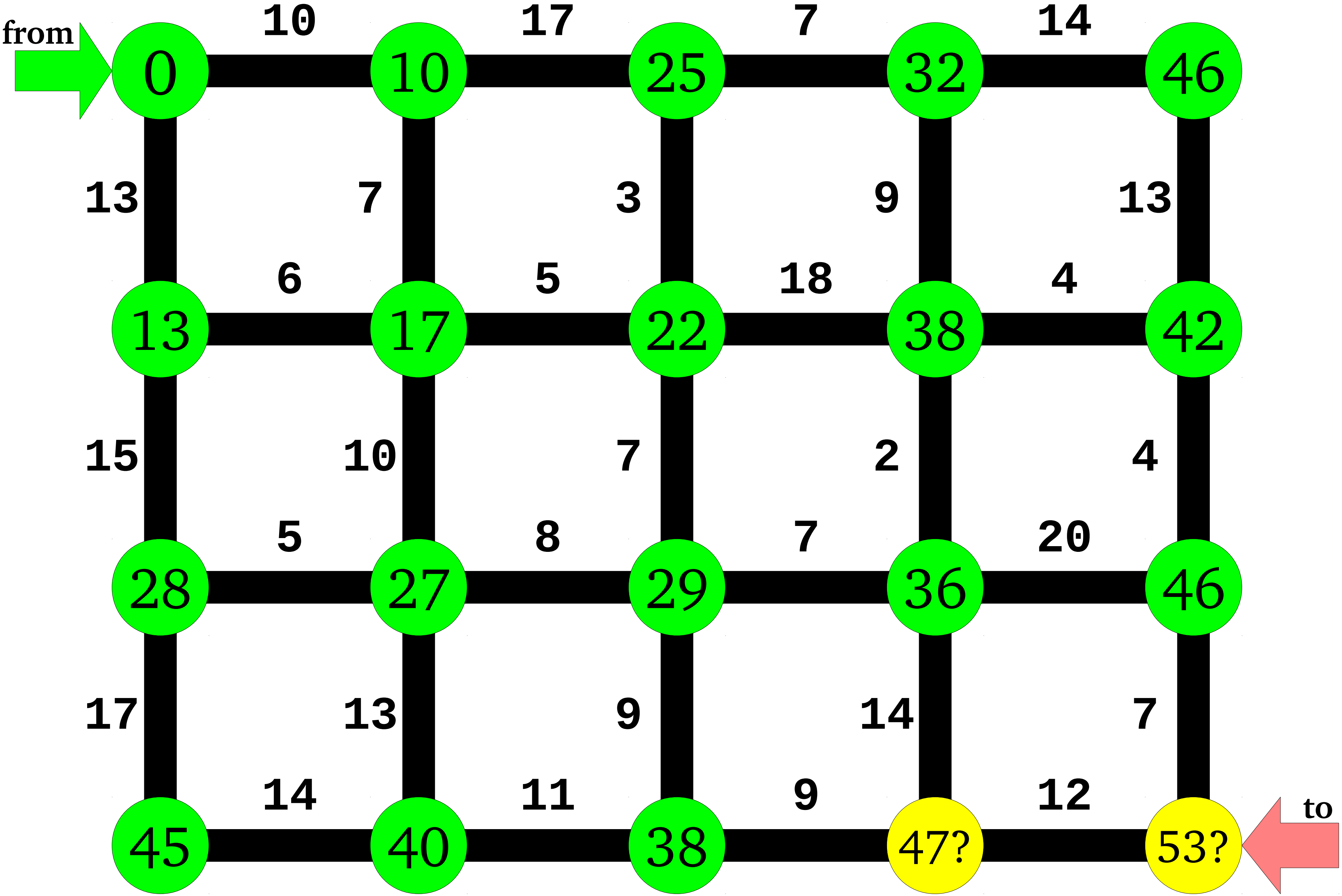

from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
38
42
15
10
7
2
4
5
8
7
20
28
27
29
36
56?
17
13
9
14
7
14
11
9
12
to
45?
40
38
47?

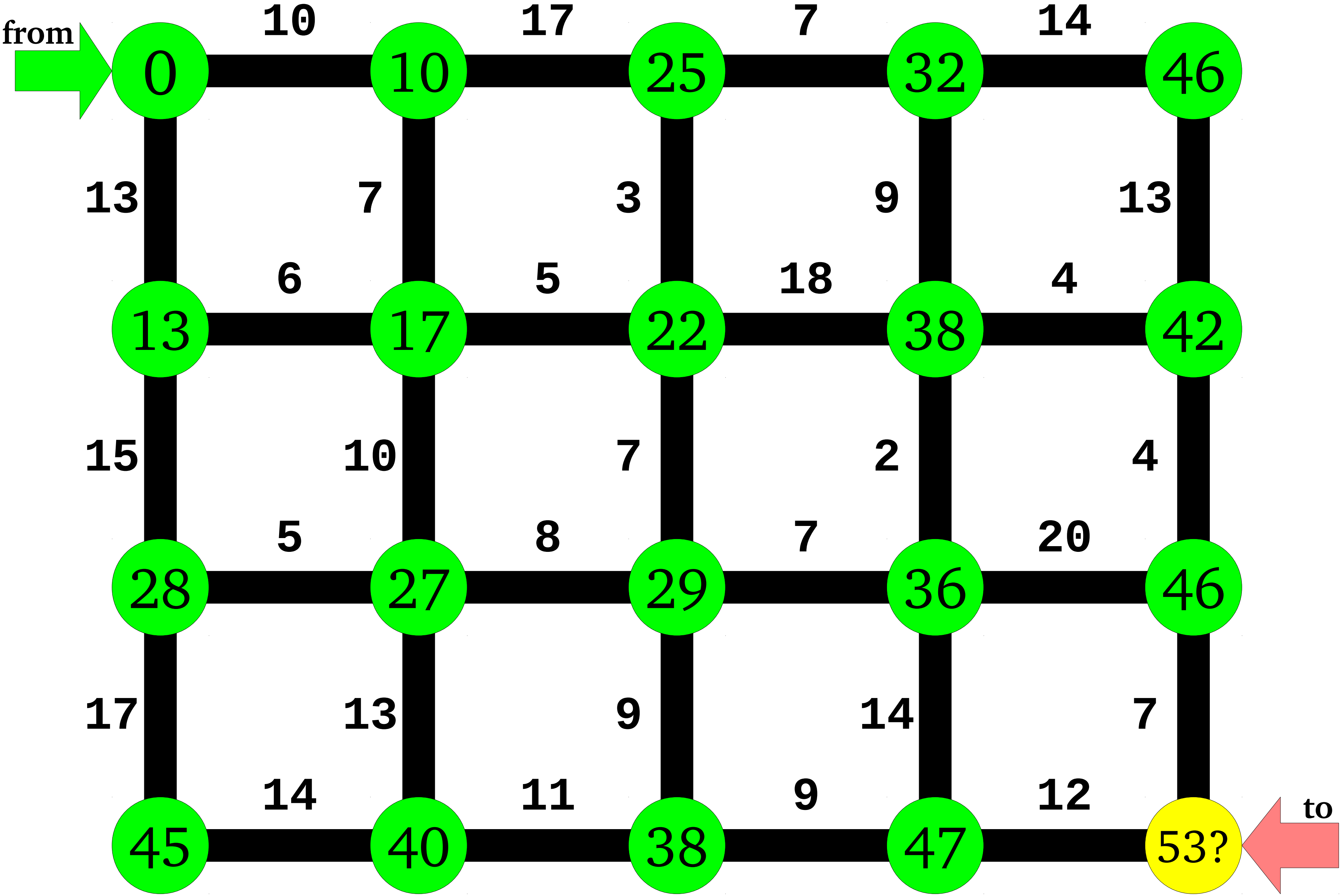

from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
38
42
15
10
7
2
4
5
8
7
20
28
27
29
36
46?
17
13
9
14
7
14
11
9
12
45?
40
38
47?
to

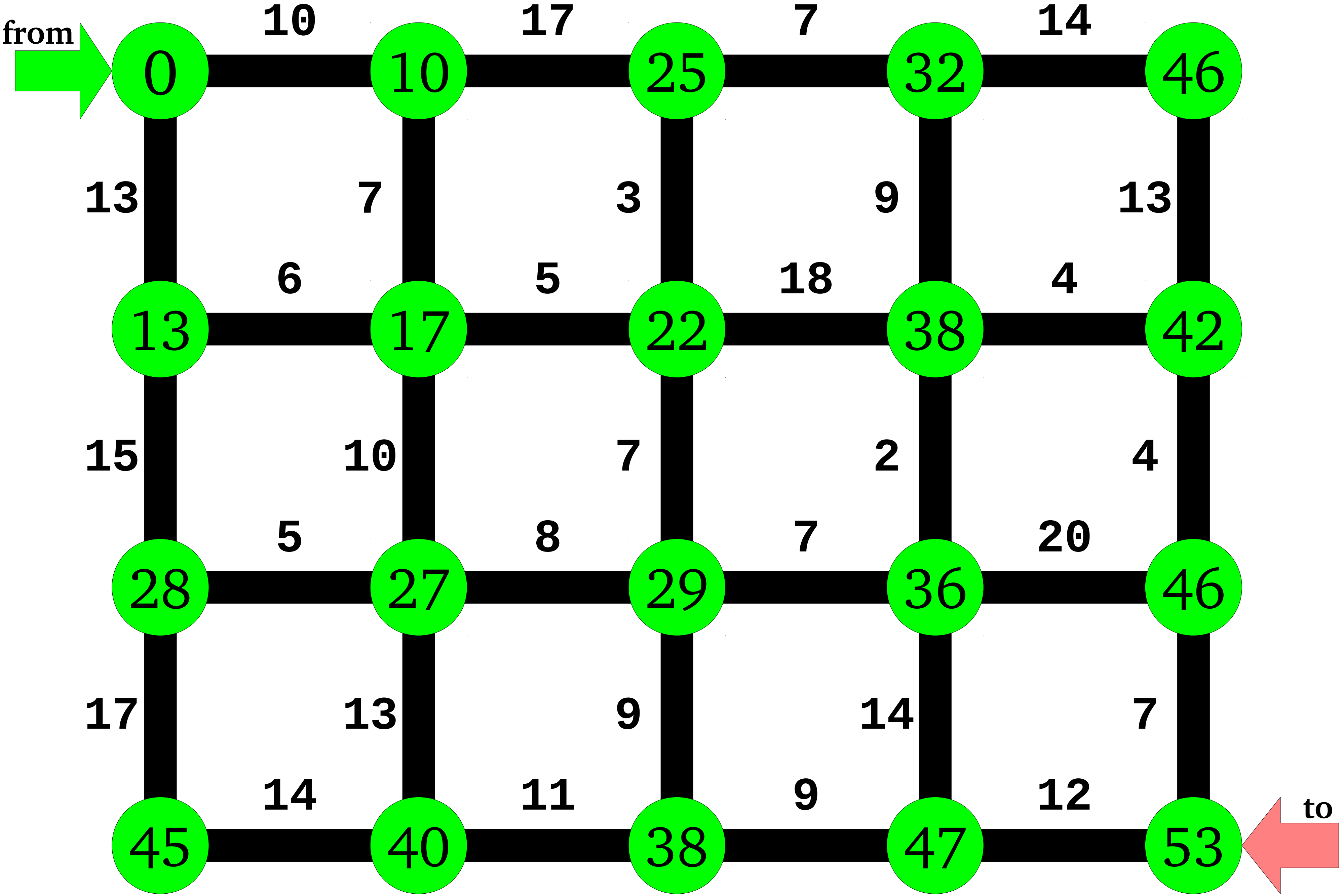

10
17
7
14
from
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
38
42
15
10
7
2
4
5
8
7
20
28
27
29
36
46?
17
13
9
14
7
14
11
9
12
to
45
40
38
47?


from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
38
42
15
10
7
2
4
5
8
7
20
28
27
29
36
46
17
13
9
14
7
14
11
9
12
to
45
40
38
47?

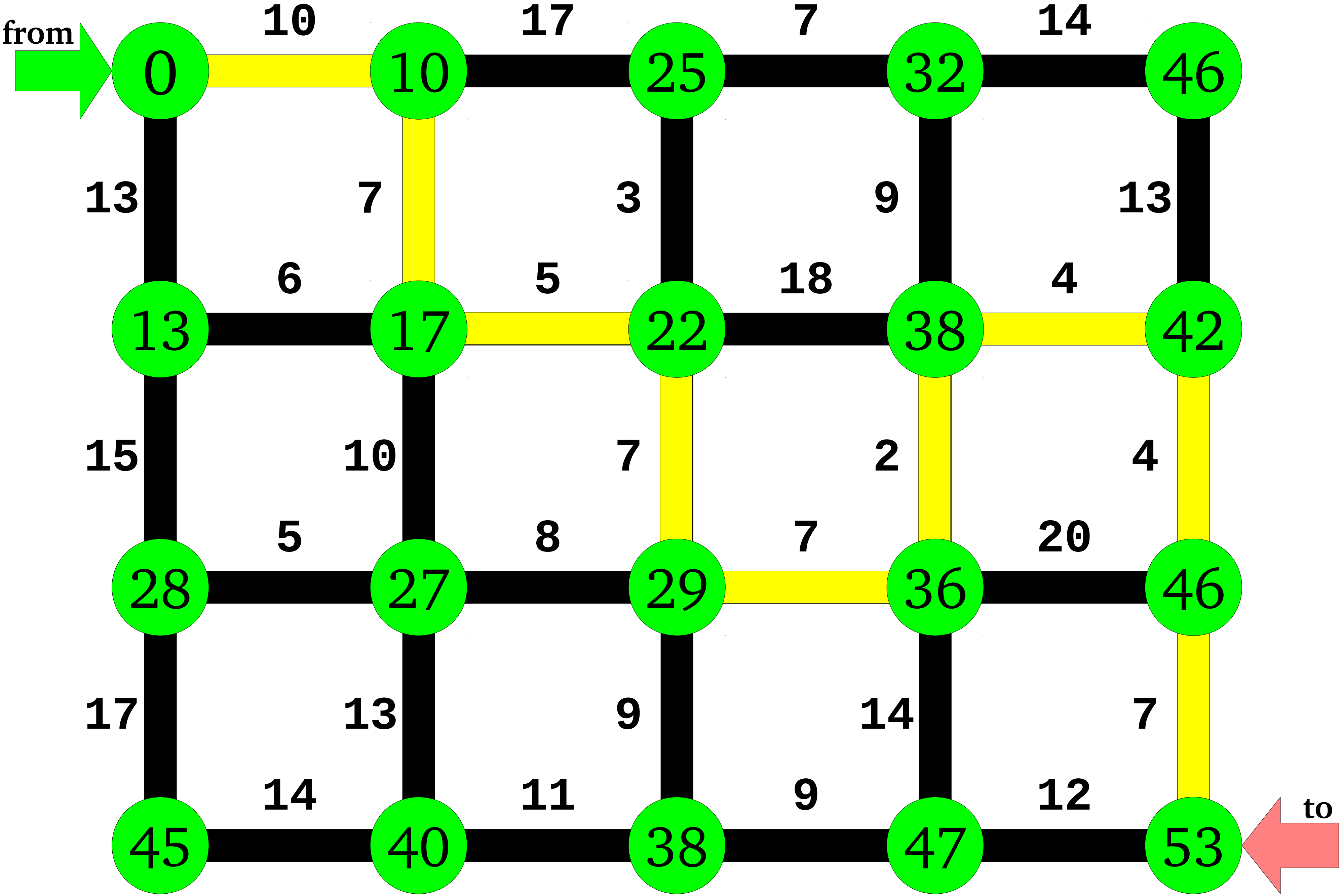

from
10
17
7
14
0
10
25
32
46?
13
7
3
9
13
6
5
18
4
13
17
22
38
42
15
10
7
2
4
5
8
7
20
28
27
29
36
46
17
13
9
14
7
14
11
9
12
to
45
40
38
47?
53?

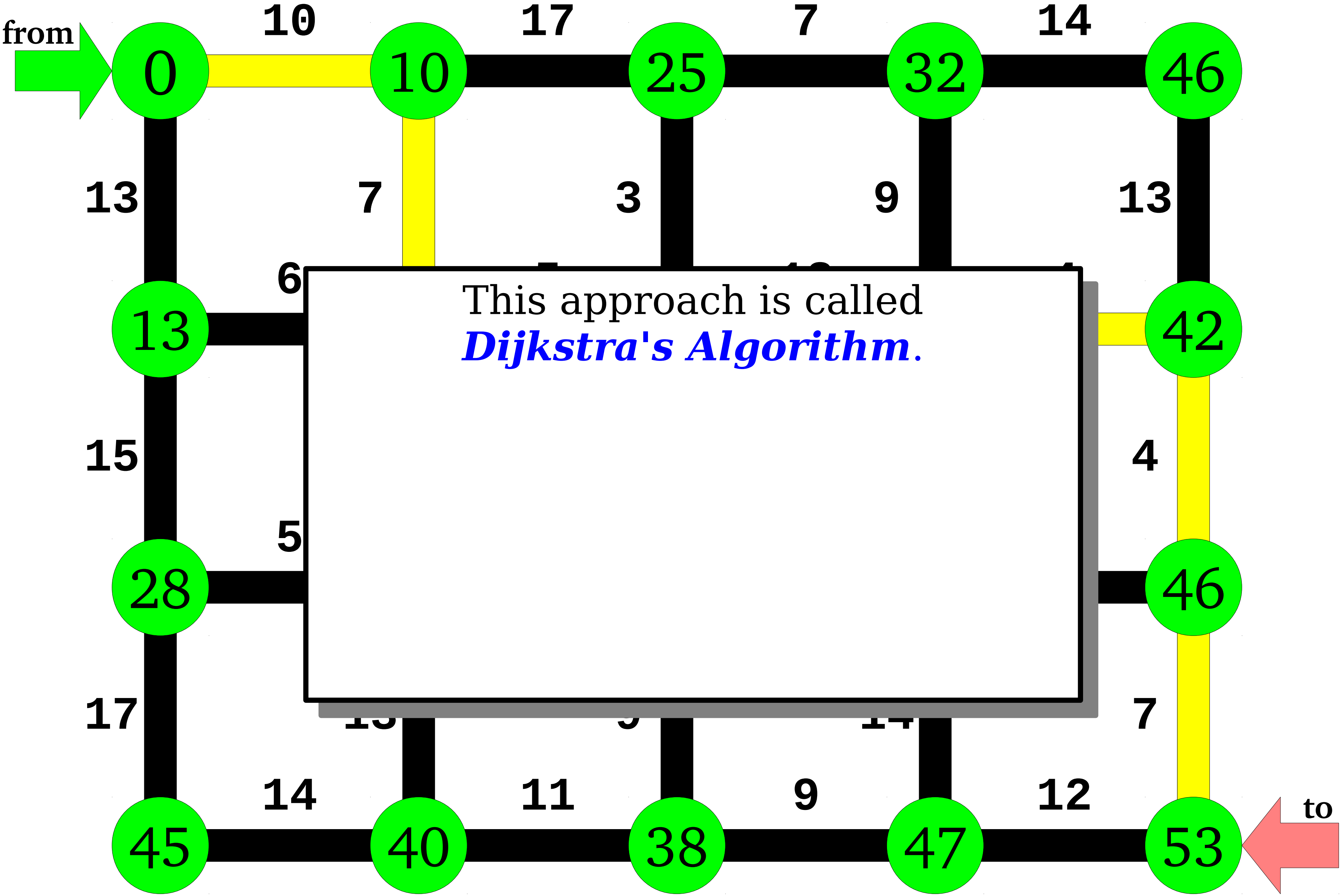

from
10
17
7
14
0
10
25
32
46
13
7
3
9
13
6
5
18
4
13
17
22
38
42
15
10
7
2
4
5
8
7
20
28
27
29
36
46
17
13
9
14
7
14
11
9
12
to
45
40
38
47?
53?


from
10
17
7
14
0
10
25
32
46
13
7
3
9
13
6
5
18
4
13
17
22
38
42
15
10
7
2
4
5
8
7
20
28
27
29
36
46
17
13
9
14
7
14
11
9
12
to
45
40
38
47
53?

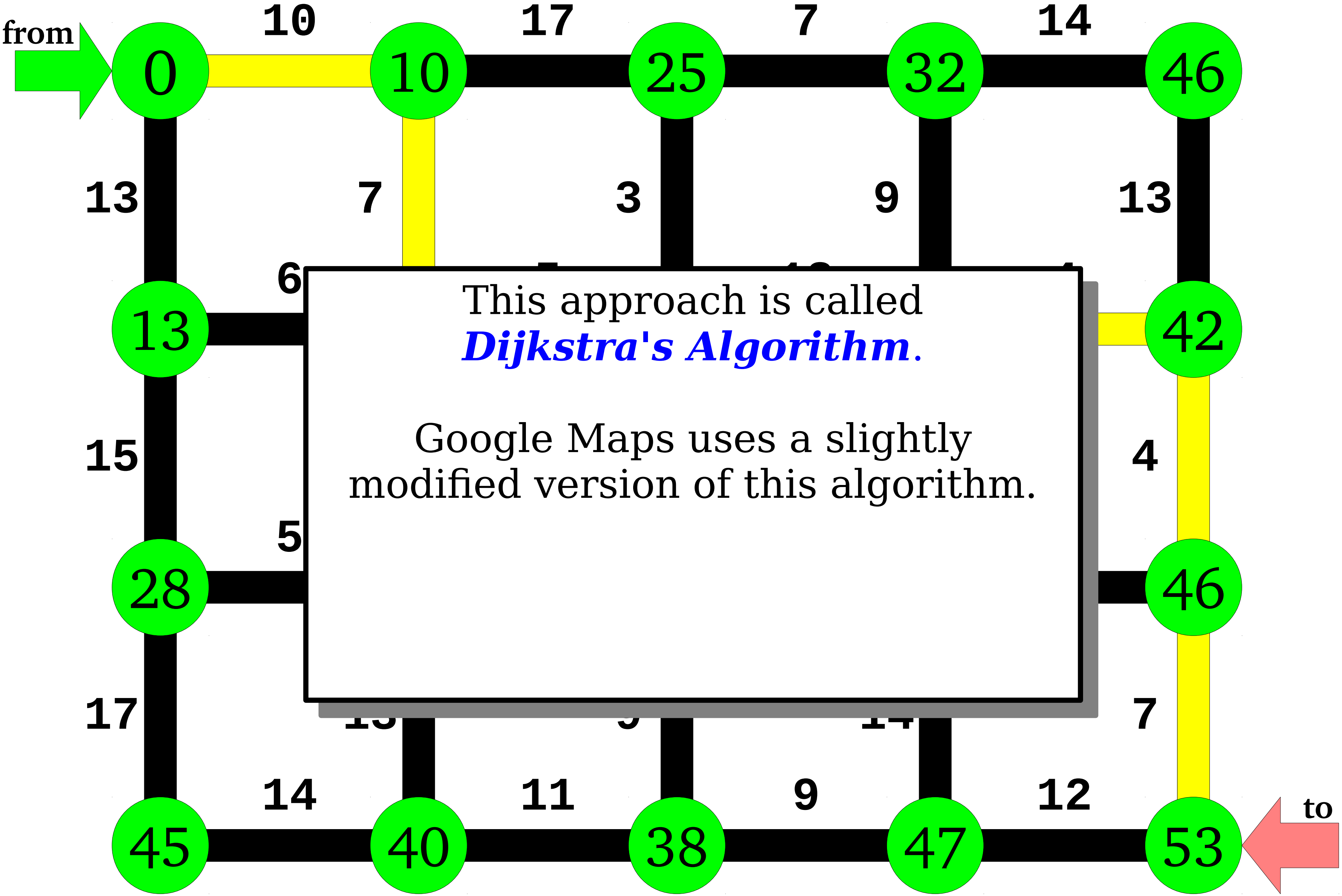


from
to
0
10
25
32
46
13
17
22
38
42
28
27
29
36
46
45
40
38
47
53
10
17
7
14
13
7
3
9
13
6
5
18
4
15
10
7
2
4
5
8
7
20
17
13
9
14
7
14
11
9
12


from
10
17
7
14
0
10
25
32
46
13
7
3
9
13
6
5
18
4
13
17
22
38
42
15
10
7
2
4
5
8
7
20
28
27
29
36
46
17
13
9
14
7
14
11
9
12
to
45
40
38
47
53

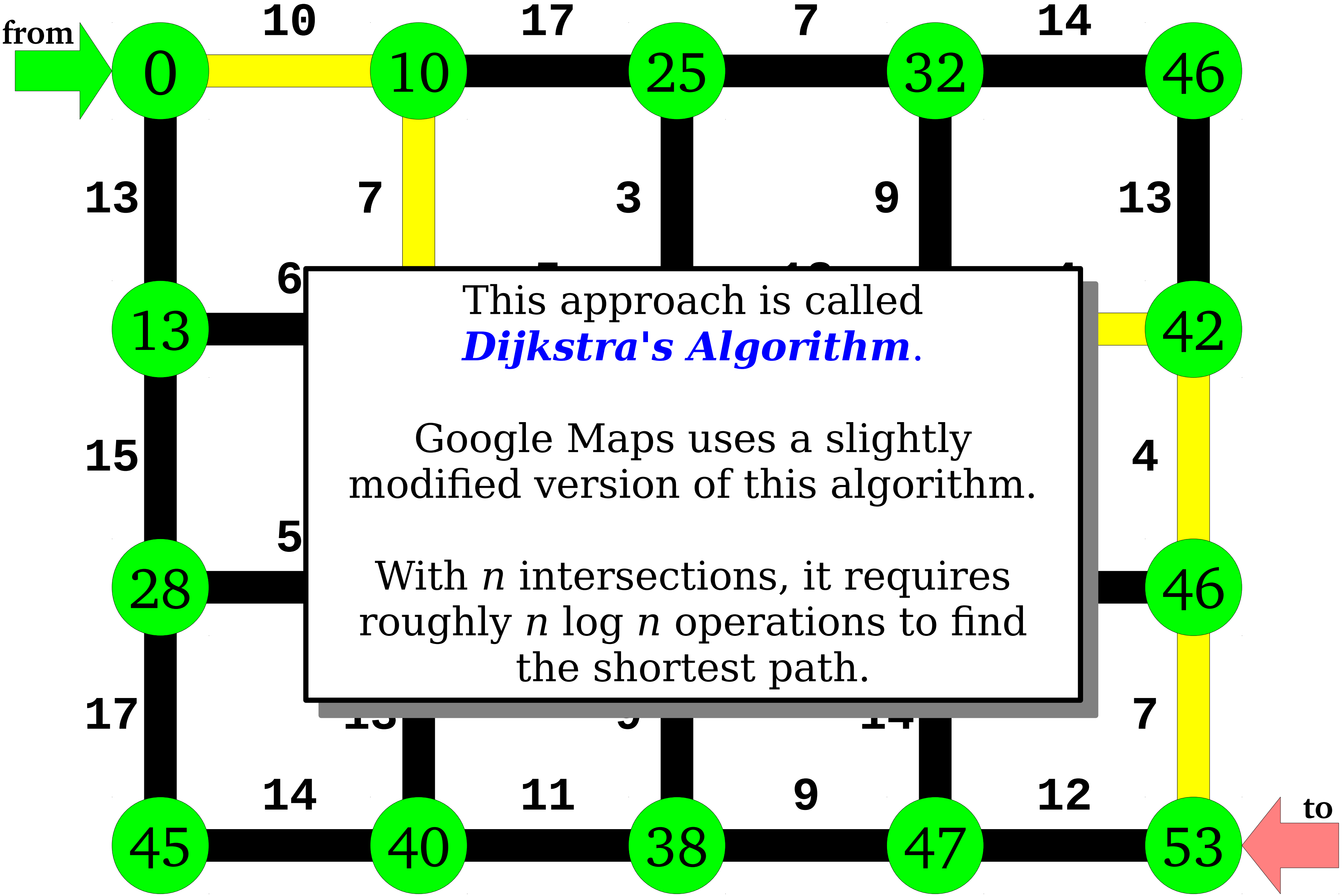


This approach is called
Dijkstra's Algorithm.



This approach is called
***Dijkstra's Algorithm*.**

Google Maps uses a slightly modified version of this algorithm.



This approach is called
Dijkstra's Algorithm**.**

Google Maps uses a slightly modified version of this algorithm.

With *n* intersections, it requires roughly *n* log *n* operations to find the shortest path.

Goals for this 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.
  - Quantitatively analyze different approaches for solving problems.

Who's Here Today?

- Aeronautics and Astronautics
- Biochemistry
- Bioengineering
- Biology
- Biomedical Informatics
- Business Administration
- Chemical Engineering
- Chemistry
- Chinese
- Civil and Environmental Engineering
- Computational and Mathematical Engineering
- Computer Science
- Creative Writing
- East Asian Studies
- Economics
- Electrical Engineering
- Energy Resources Engineering
- Engineering
- Environment and Resources
- Feminism, Gender, and Sexuality Studies
- Film and Media Studies
- German Studies
- Human Biology
- Immunology
- International Policy Studies
- Law
- Management Science and Engineering
- Materials Science and Engineering
- Mathematical and Computational Sciences
- Mechanical Engineering
- Medicine
- Music
- Petroleum Engineering
- Physics
- Political Science
- Psychology
- Public Policy
- Science, Technology, and Society
- Statistics
- Stem Cell Biology and Regenerative Medicine
- Symbolic Systems
- Theater and Performing Studies
- ***Undeclared!***

One more detail…

C++

What is C++?

- C++ is a widely used programming language used to design all sorts of systems that are
  - complex, but
  - need to run fast.
- The syntax of Java was influenced by the syntax of C++.
- There are many features of C++ that aren’t present in Java, and those features make it an attractive language for use in CS106B.
- C++ is a ***huge*** language that’s undergone many revisions (it was invented in 1983; most recent version is C++14) and we won’t be covering it in full depth. Take CS106L or CS110 for more!

```cpp
/* File: hello-world.cpp
 *
 * A canonical Hello, world! program
 * in C++.
 */

#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;
}
```

```cpp
/* File: retain-evens.cpp
 *
 * A program to filter out odd numbers from a list.
 */
#include <iostream>
#include "vector.h"
using namespace std;

Vector<int> retainEvens(Vector<int> values) {
    Vector<int> result;
    for (int i = 0; i < values.size(); i++) {
        if (values[i] % 2 == 0)
            result += values[i];
    }
    return result;
}

int main() {
    Vector<int> values;
    values += 1, 2, 3, 4, 5;

    Vector<int> processed = retainEvens(values);

    for (int elem: processed) {
        cout << elem << endl;
    }
}
```

Your Action Items

- ***Assignment 0: Welcome to CS106B*** is due this Friday at the start of class (11:30AM).
  - Starter files and assignment handout are up on the course website.
  - No programming involved, but you'll need to get your development environment set up.
  - There's a bunch of documentation up on the course website. Please feel free to reach out to us if there's anything we can do to help out!
- Some of the later assignments can be done in pairs. You may want to start thinking about who you'd like to work with, since you'll need to register for the same section as the person you'll be working with.

Next Time

- ***Welcome to C++!***
  - Defining functions.
  - Reference parameters.
  - Introduction to recursion.