PDST

Professional Development Service for Teachers

An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

An Roinn Oideachais agus Scileanna
Department of Education and Skills

National Workshop 4

LEAVING CERTIFICATE
COMPUTER SCIENCE

Schedule

Session 1	Algorithms II
11.00 – 11.30	Tea/Coffee
Session 2	Computer Systems II
13.00 – 14.00	Lunch
Session 3	ALT3 - Project Design and Presentation CWA Reporting Requirements

Key Messages for NW4

There are many ways to use the LCCS specification.

The study of Computers and Society is one of the overarching principle of LCCS

LCCS can be mediated through a constructivist pedagogical approach.

ALTs

ALTs provide an opportunity to teach theoretical aspects of LCCS.

Critical reflection will be a central component of the student experience and the LCCS teacher's PD journey.

Digital technologies can be used to enhance collaboration, learning and reflection.

PDST

Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Session 1

Algorithms II

Overview of the Session

Section 1	Unconscious bias and algorithmic bias
Section 2	Introduction to Algorithms (revisited)
Section 3	Stretch break
Section 4	Sorting algorithms
Section 5	Binary search

By the end of this session participants will have:

- participated in an interactive dialogue on unconscious bias and **algorithmic bias**.
- reflected on the **definition** and **characteristics** of algorithms, as well as the ubiquitous nature of algorithms in today's society.
- developed their conceptual understanding of a variety of **sorting algorithms**.
- participated in an activity to **facilitate** the **effective learning** of algorithms in their own classrooms.
- developed their understanding of the **binary search** algorithm.

PDST

Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Section I

Unconscious Bias & Algorithmic Bias

A builder, leaning out of the van, shouts “nice legs” to a nurse passing by. The same nurse arrives at work, and casually mentions this to a senior doctor. The doctor said,” I’d never say that”. The doctor has two grown up children who are 22 and 30. They get on very well. One is a Sergeant in the Army; the other is training to be a beauty therapist. The doctor divorced last year and is currently dating someone else.

Mentimeter

URL = www.menti.com

Code = 3607 2682

Answer the questions based on the story from the previous slide

Menti Code = 3607 2682

A builder, leaning out of the van, shouts “nice legs” to a nurse passing by. The same nurse arrives at work, and casually mentions this to a senior doctor. The doctor said,” I’d never say that”. The doctor has two grown up children who are 22 and 30. They get on very well. One is a Sergeant in the Army; the other is training to be a beauty therapist. The doctor divorced last year and is currently dating someone else.

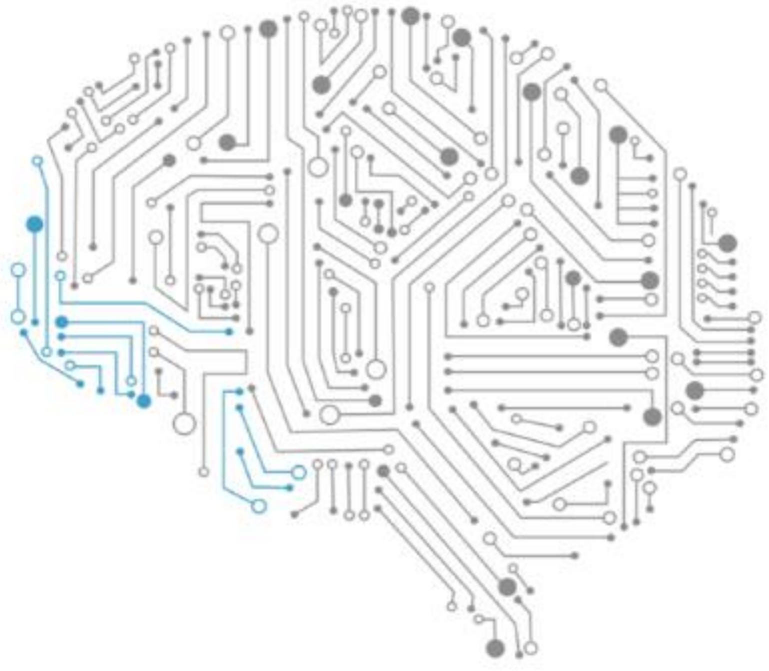
Answers to the quiz…

	True		False		Don't know
There was at least one man in the van					✓
Not every man mentioned would shout 'nice legs'					✓
The doctor is no longer living with his wife					✓
The doctor has a new girlfriend					✓
The doctor's son is in the army					✓
The youngest child is training to be a beauty therapist					✓
At some point a man spoke to a woman					✓
At least two of the people mentioned are men					✓
A woman was shouted at					✓

What is Unconscious Bias?

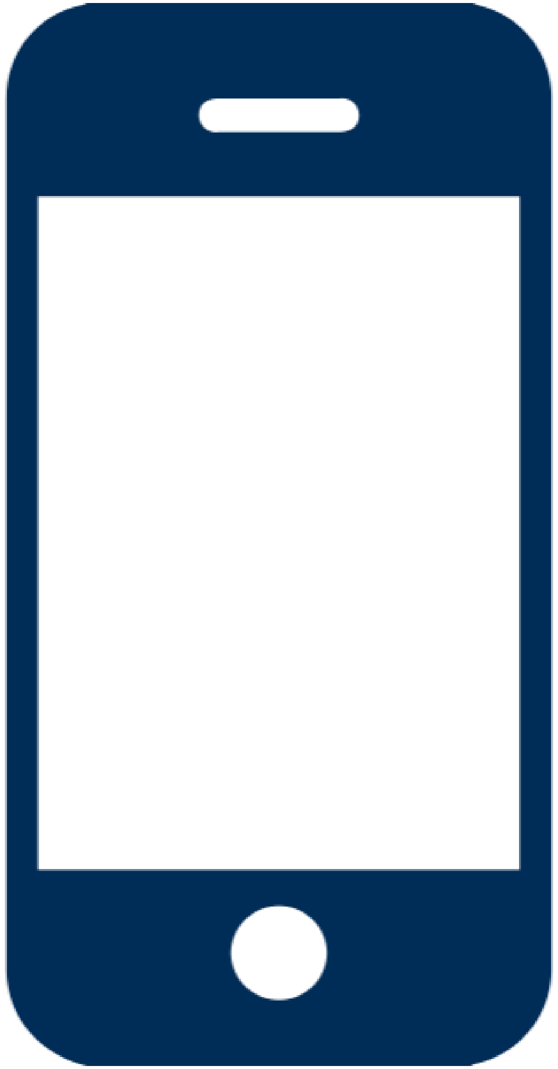
- Natural
- Rapid categorization of people
- Created by social influence
- Unintentional
- Used by everyone
- Most likely to be acted on when we are stressed or tired
- A bad thinking habit

What does Unconscious Bias have to do with Algorithms?



“some of the biggest problems in the industry aren’t technical – they’re people (egos etc.) …. diversity creates better solutions and opportunities because of wider experience set, perspectives and **less bias ...**

James Whelton, CoderDojo Co-Founder



Created by i cons
from Noun Project

Algorithmic bias

“The nurse said that…”

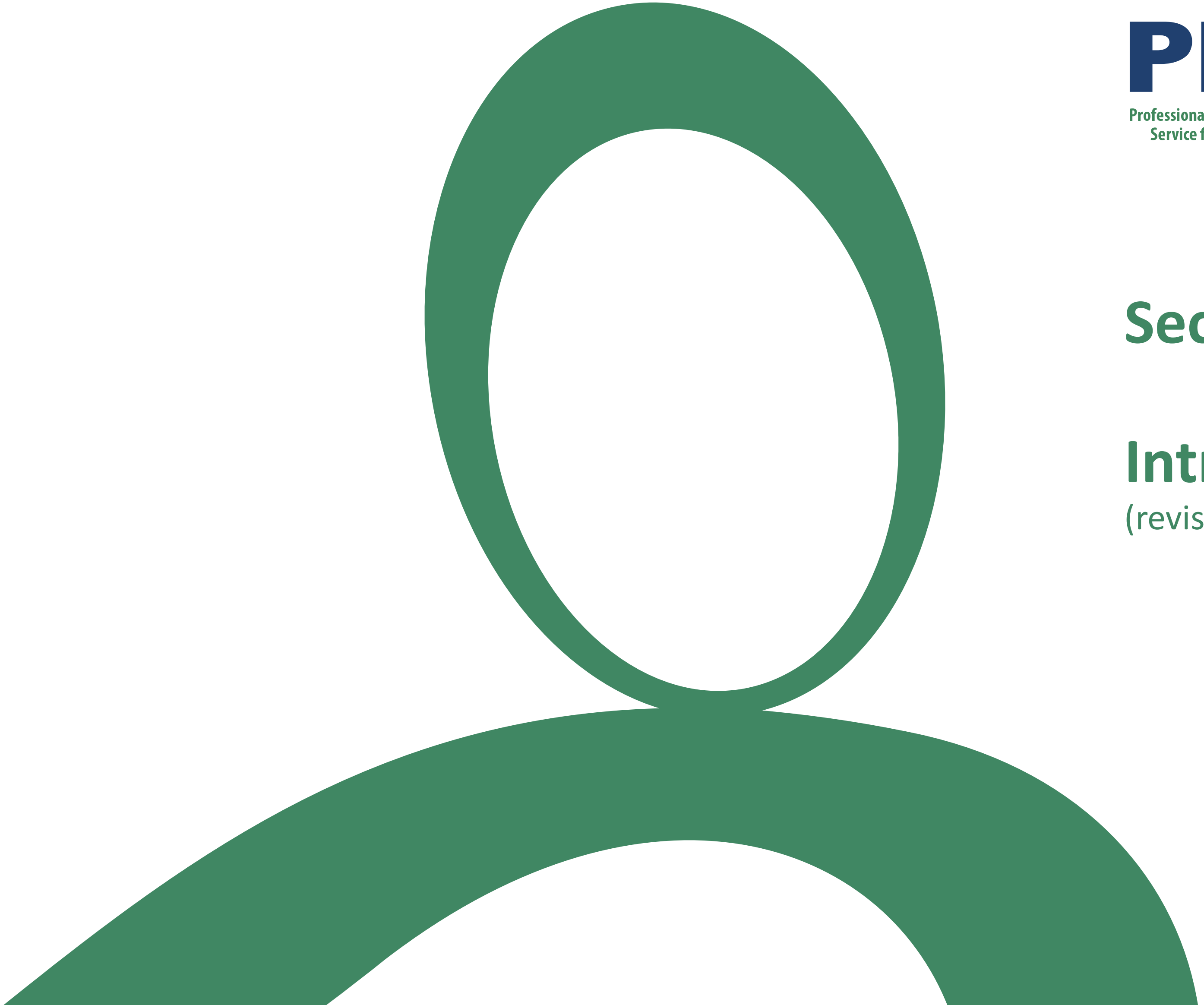
“The pilot said that…”

What does Unconscious Bias have to do with YOU?

Personal Reflection

What are my values as a teacher?

PDST
Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí



Section II

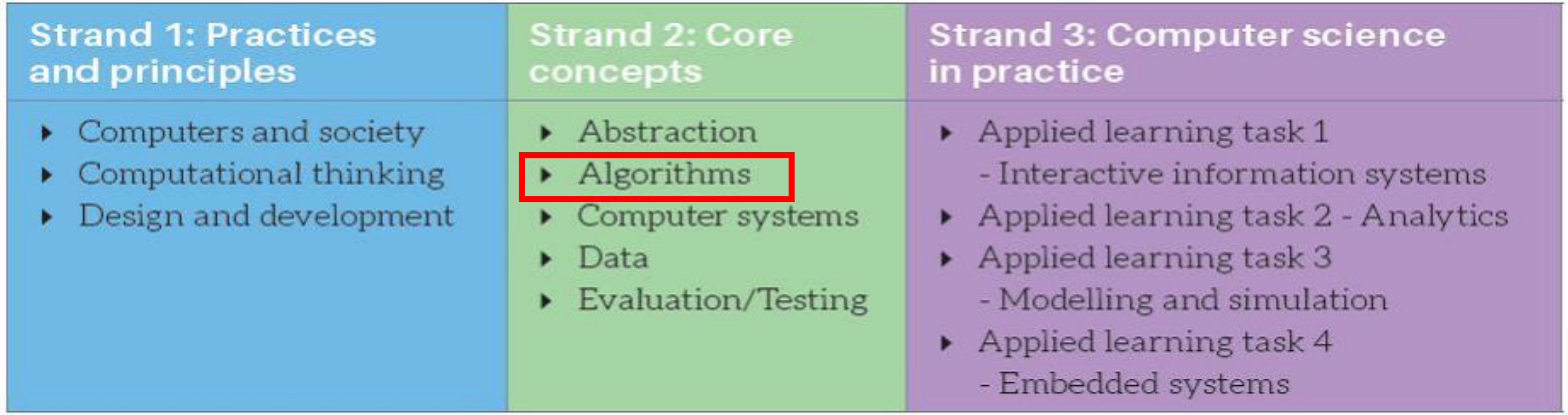
Introduction to Algorithms

(revisited)

Algorithms and the Specification

“Computer science is the study of computers and algorithmic processes. Leaving Certificate Computer Science includes how programming and computational thinking can be applied to the solution of problems, and how computing technology impacts the world around us. “

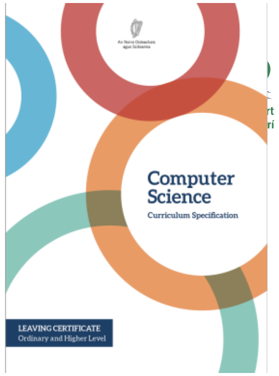
NCCA Curriculum specification, Page 1



Strand 1: Practices and principles	Strand 2: Core concepts	Strand 3: Computer science in practice
▸ Computers and society ▸ Computational thinking ▸ Design and development	▸ Abstraction ▸ Algorithms ▸ Computer systems ▸ Data ▸ Evaluation/Testing	▸ Applied learning task 1 - Interactive information systems ▸ Applied learning task 2 - Analytics ▸ Applied learning task 3 - Modelling and simulation ▸ Applied learning task 4 - Embedded systems

NCCA Curriculum specification, Page 11

LCCS Learning Outcomes



Computer Science

Curriculum Specification

LEAVING CERTIFICATE

Ordinary and Higher Level

2.5 use pseudo code to outline the functionality of an algorithm

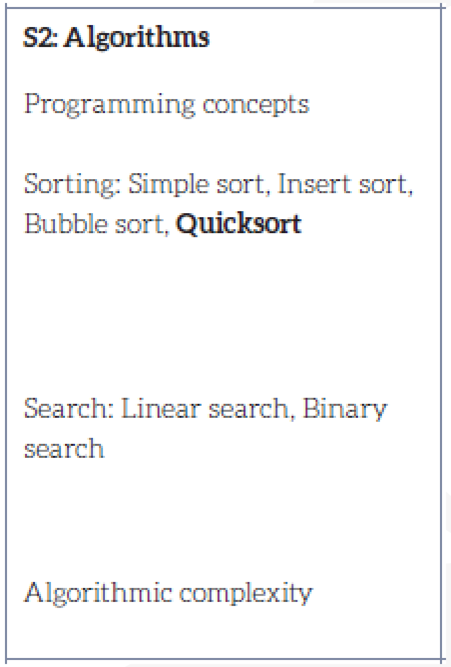
2.6 construct algorithms using appropriate sequences, selections/conditionals, loops and operators to solve a range of problems, to fulfil a specific requirement

2.7 implement algorithms using a programming language to solve a range of problems

2.8 apply basic search and sorting algorithms and describe the limitations and advantages of each algorithm

2.9 assemble existing algorithms or create new ones that use functions (**including recursive**), procedures, and modules

2.10 explain the common measures of algorithmic efficiency using any algorithms studied



S2: Algorithms

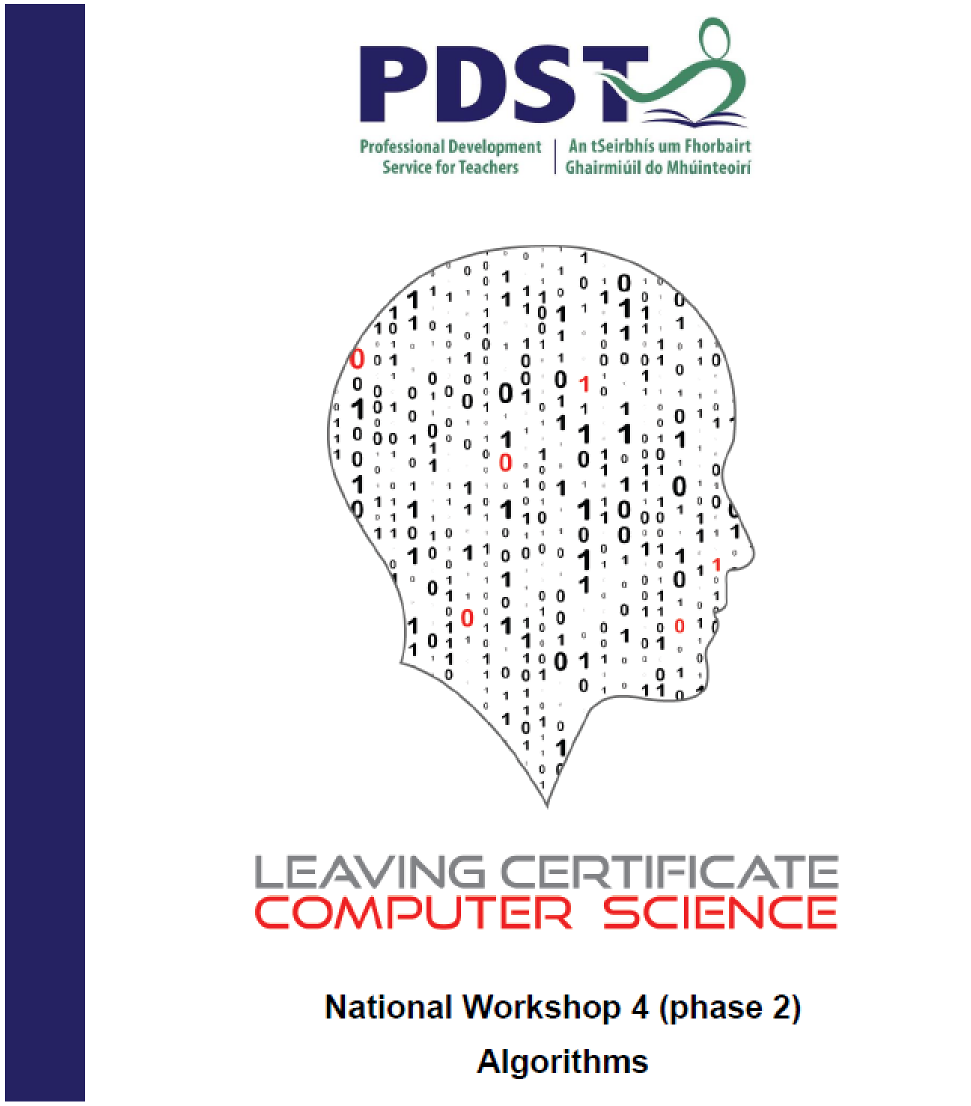
Programming concepts

Sorting: Simple sort, Insert sort, Bubble sort, **Quicksort**

Search: Linear search, Binary search

Algorithmic complexity

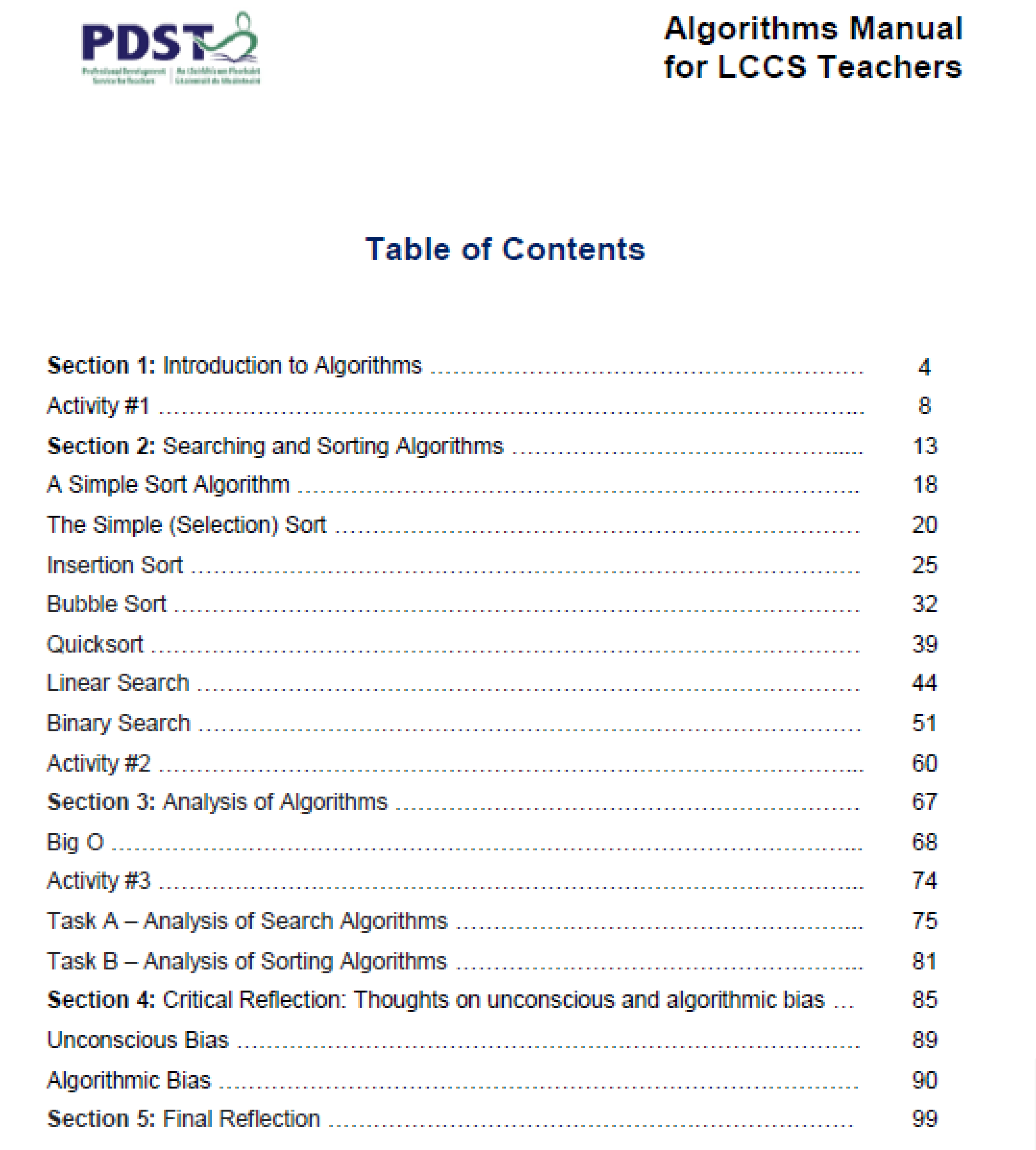
See also learning outcomes 1.6, 1.7 1.14, 1.22, 2.3, 3.4 and 3.7 … plus others



LEAVING CERTIFICATE
COMPUTER SCIENCE

National Workshop 4 (phase 2)

Algorithms



Algorithms Manual
for LCCS Teachers

Table of Contents

COMPSCI.IE

https://www.compsci.ie/uploads/resources/35873/35641.pdf

What is an algorithm?

> “An algorithm is a set of rules for getting a specific output from a specific input. Each step must be so precisely defined that it can be translated into computer language and executed by machine”

Source: Knuth, D The Art of Computer Programming (Vol. 1, Fundamental Algorithms, 3rd ed.)

Donald Knuth

According to Knuth, an algorithm has five important features:

Finiteness	An algorithm must always terminate after a finite number of steps.
Definiteness	Each step must be precisely defined.
Input	An algorithm has zero or more inputs.
Output	An algorithm has one or more outputs, which have a specified relation to the inputs.
Effectiveness	All operations to be performed must be sufficiently basic that they can in principle be done exactly and in finite length of time by someone using pencil and paper.

What is an algorithm?

A step-by-step procedure for solving a problem or accomplishing some end especially by a computer

Merriam-Webster

- ✓ A sequence of instructions
- ✓ A way of capturing intelligence and sharing it with others
- ✓ Provide general solutions to problems
- ✓ Some problems are so hard that they cannot be solved by algorithms (Computability)
- ✓ Can be expressed in a variety of different ways
- ✓ Common elements include input, processing, output
- ✓ Close relationship between algorithms and data structures
- ✓ Essential features are correctness and effectiveness
- ✓ Rule-based algorithms vs. Machine learning algorithms (AI)

Section III

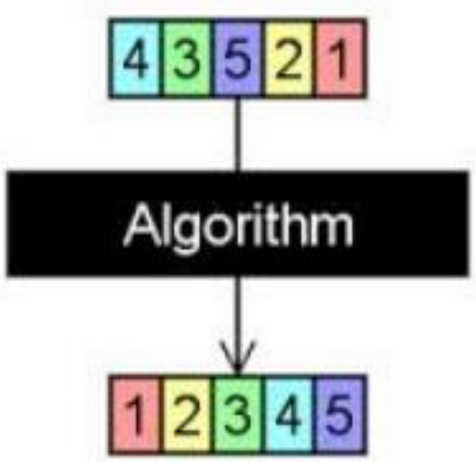
Sorting Algorithms

Sorting Algorithms

An algorithm that maps the following input/output pair is called a sorting algorithm:

Input: A list (or array), L, that contains n orderable elements: $\mathrm{L}[0, 1, \ \dots \ , \mathrm{n} - 1]$.

Output: A sorted permutation of L, called S, such that $S[0] \leq S[1] \leq \cdots \leq S[\mathrm{n} - 1]$.



Simple (selection) Sort

Insertion Sort

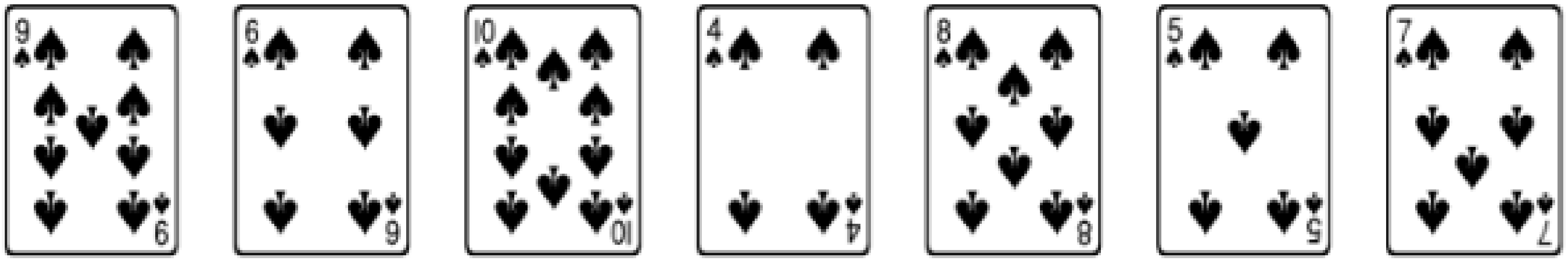
Bubble Sort

Quicksort

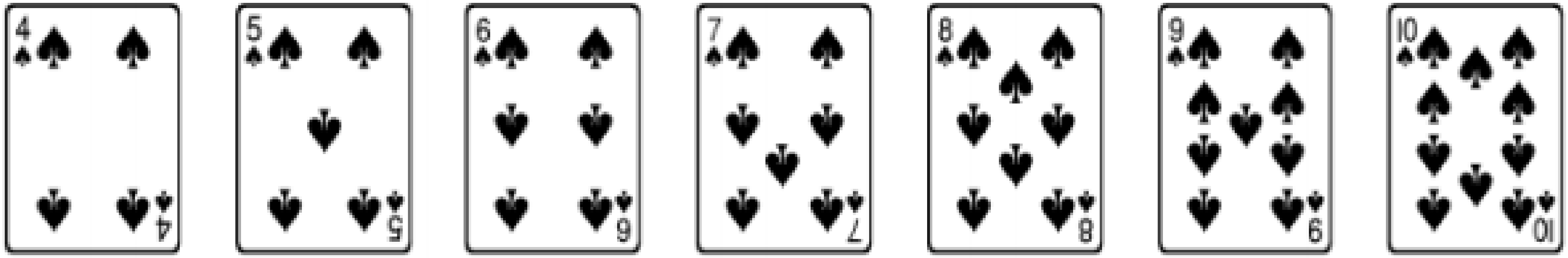
Group Activity / Breakout

Simple Sort Demonstration

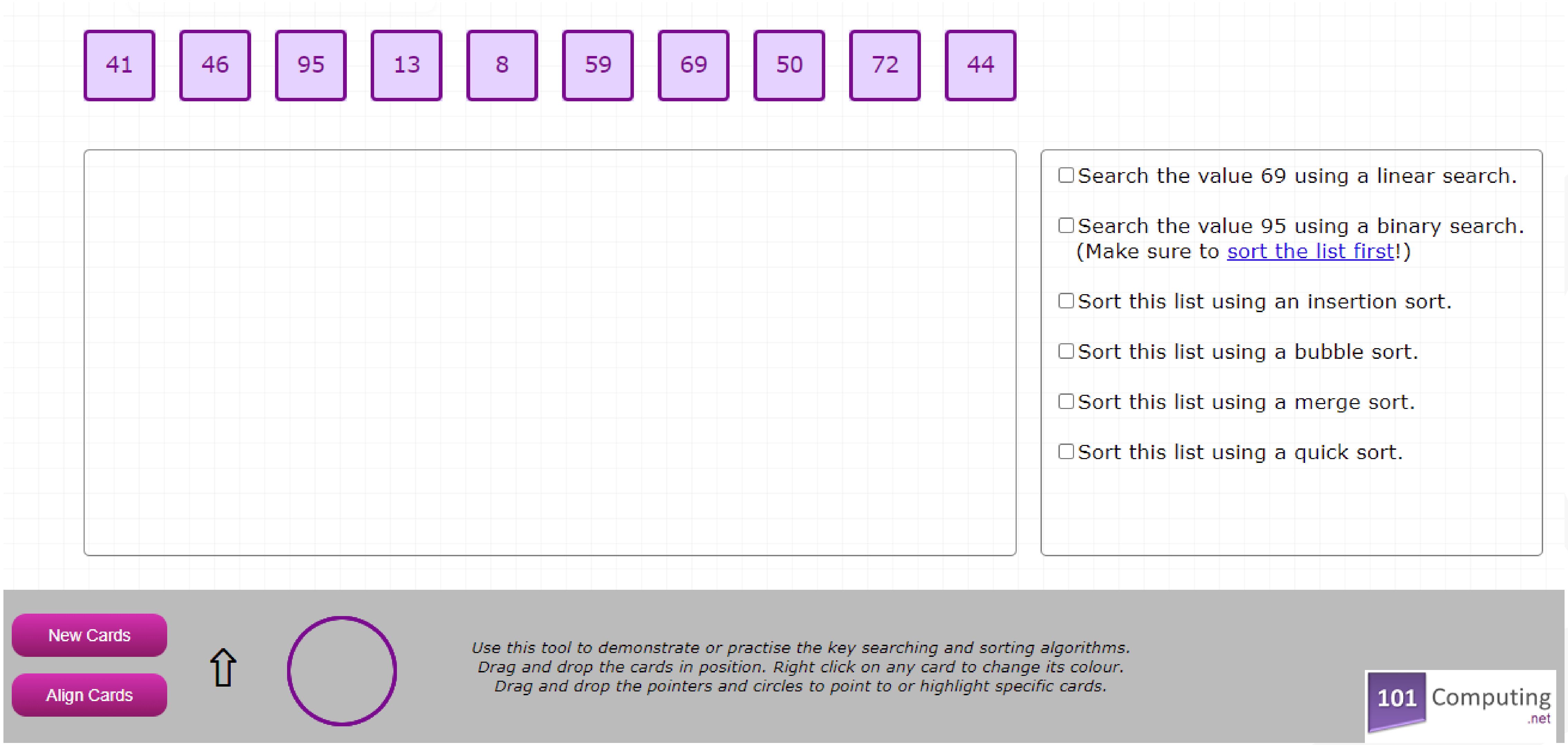
Input



Required Output:



Simple Sort Demonstration



41 46 95 13 8 59 69 50 72 44

□ Search the value 69 using a linear search.

□ Search the value 95 using a binary search. (Make sure to sort the list first!)

□ Sort this list using an insertion sort.

□ Sort this list using a bubble sort.

□ Sort this list using a merge sort.

□ Sort this list using a quick sort.

Use this tool to demonstrate or practise the key searching and sorting algorithms.
Drag and drop the cards in position. Right click on any card to change its colour.
Drag and drop the pointers and circles to point to or highlight specific cards.

https://www.101computing.net/card-sort/

Group Activity / Breakout

Instructions :

1. Individuals read the algorithm provided and develop their own undserstanding (5-10mins.)

2. Each group then discusses and agrees a common understanding of their assigned algorithm

3. Groups prepare a demonstration/explanation which they will use to teach others (after the breakout)
(https://www.101computing.net/card-sort/)

Appoint a chair, a timekeeper, a notetaker and a spokesperson

Simple (selection) Sort

Insertion Sort

Bubble Sort

Groups and Topics

Groups 1,2,3	Simple (Selection) Sort	Pages 20-24
Groups 4,5,6	Insertion Sort	Pages 25-31
Groups 7,8,9	Bubble Sort	Pages 32-39

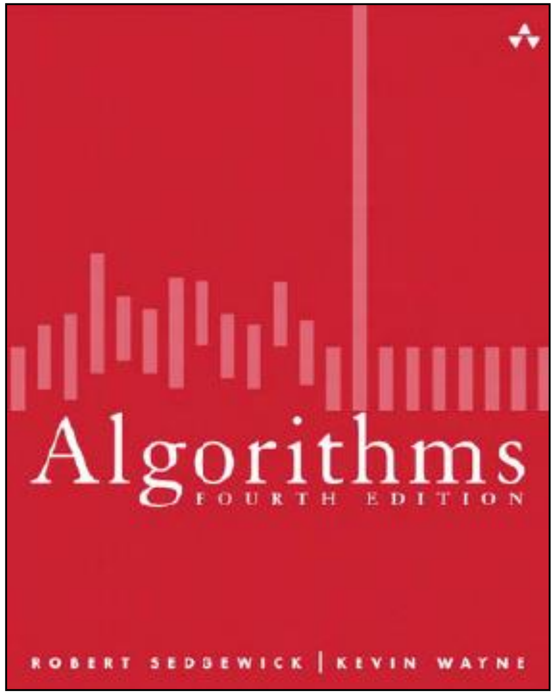
Appoint a chair, a timekeeper, a notetaker and a spokesperson

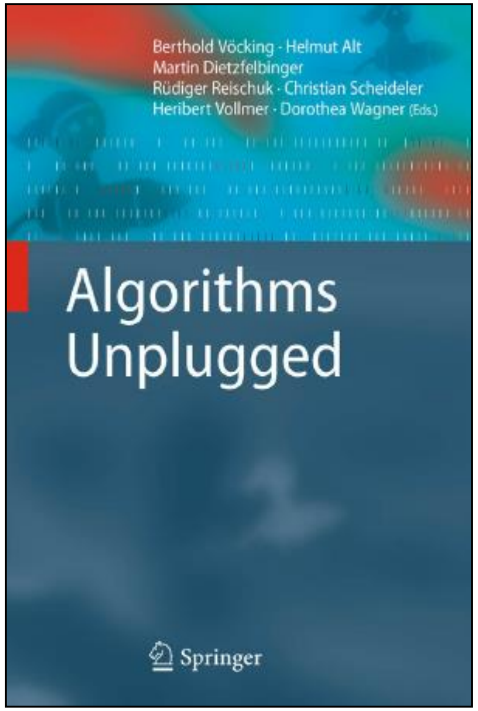
Group Feedback to others

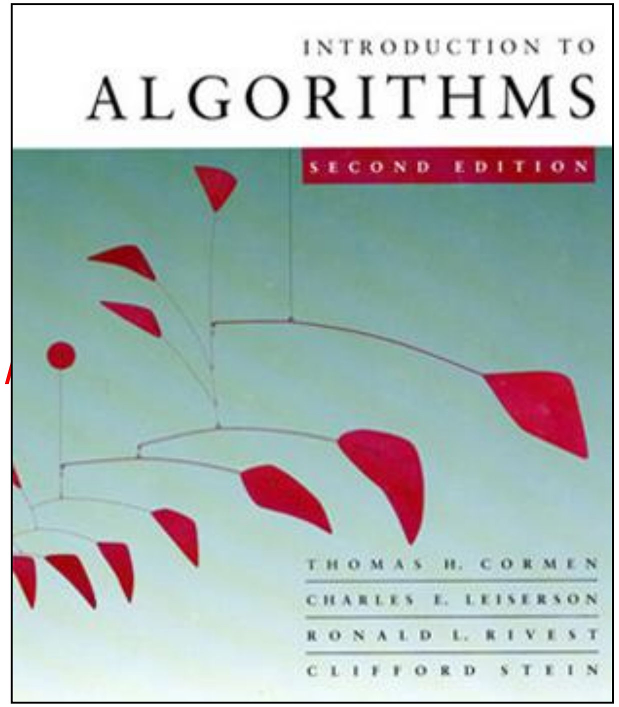
Each Breakout Room explains their algorithm to others

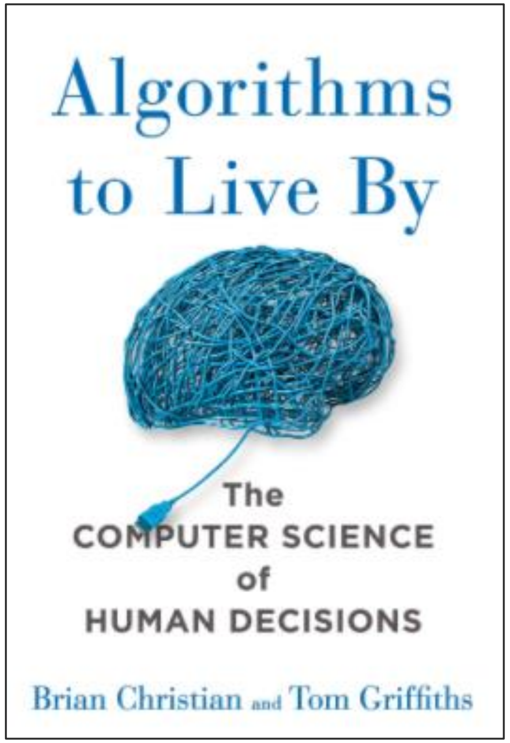
5 minute movement break

Resources




Berthold Vöcking · Helmut Alt
Martin Dietzfelbinger
Rüdiger Reischuk · Christian Scheideler
Heribert Vollmer · Dorothea Wagner (Eds.)
Algorithms Unplugged
Springer





COMPSCI.IE **https://www.compsci.ie/uploads/resources/35873/35641.pdf**

COMPSCI.IE Slides to accompany Algorithms Manual for LCCS Teachers

Section IV

Binary Search – quick intro

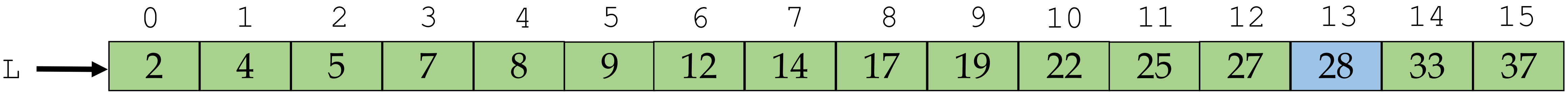
Input:

A list L and a search argument (target value) of 28

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L →	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Input:

A list L and a search argument (target value) of 28



L →

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Required Output:

If the argument is found in L, its index

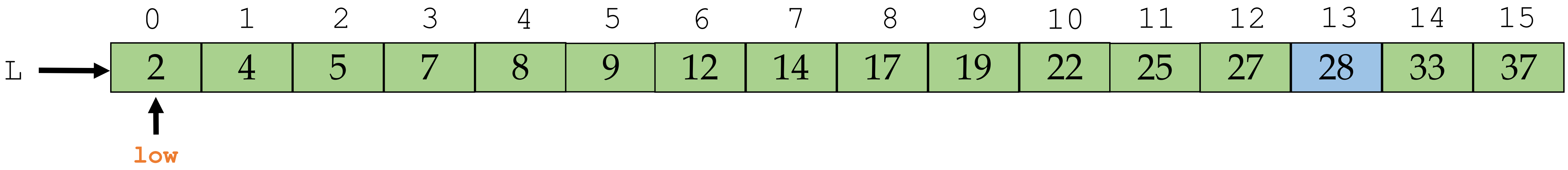
If the argument is not found in L, the length of the L (i.e. 16)

Binary Search Algorithm

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L →	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Pseudo-code: (target value is 28)

Binary Search Algorithm

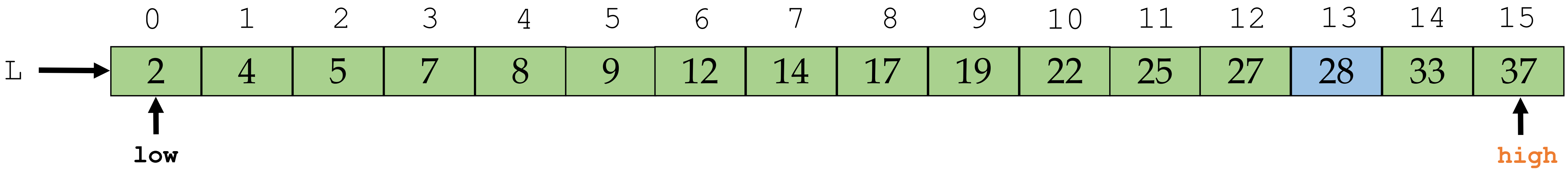


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Pseudo-code: (target value is 28)

```
1. Set low = 0
```

Binary Search Algorithm

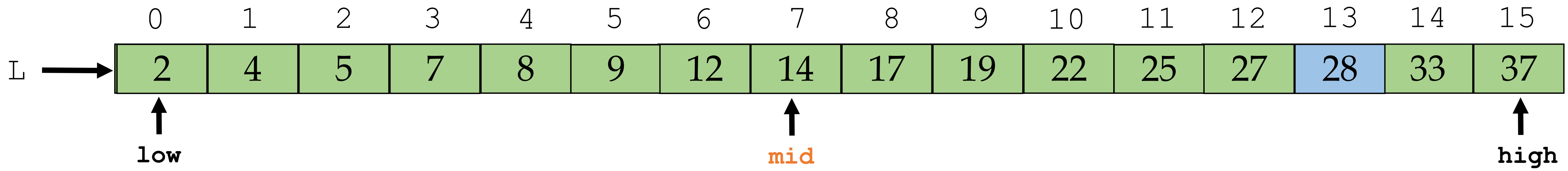


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
```

Binary Search Algorithm

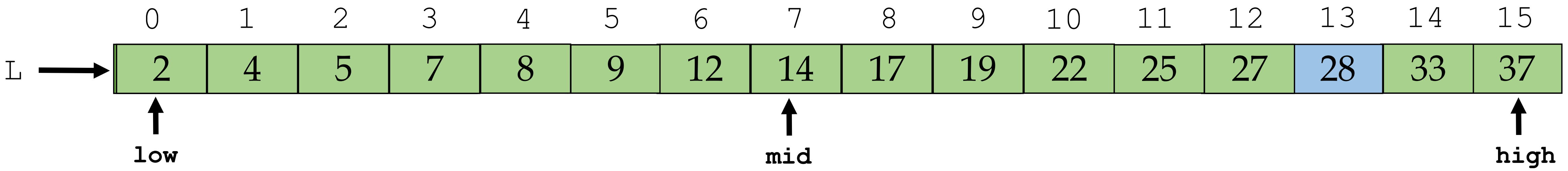


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37
	low							mid								high

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{\text{low+high}}{2}$, rounded down to an integer
```

Binary Search Algorithm

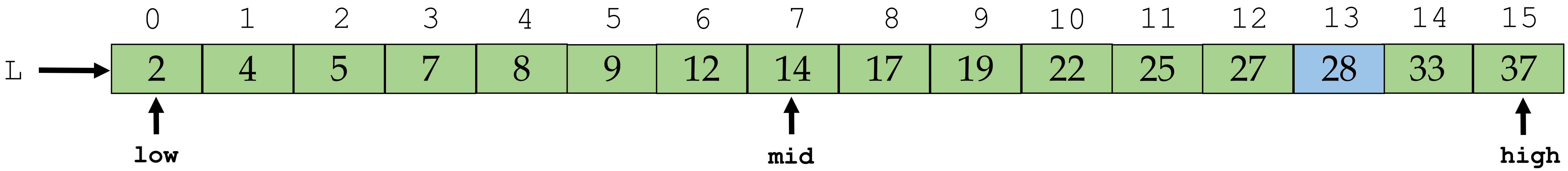


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{\text{low+high}}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
           Return mid
   Else If the value at the mid position is less than the target value
           Set low = mid + 1
   Else If the value at the mid position is greater than the target value
           Set high = mid - 1
```

Binary Search Algorithm

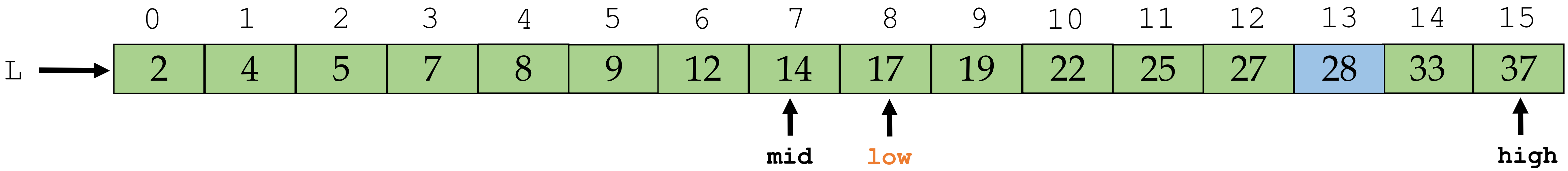


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L →	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37
	low							mid								high

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{\text{low+high}}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
           Return mid
   Else If the value at the mid position is less than the target value
           Set low = mid + 1
   Else If the value at the mid position is greater than the target value
           Set high = mid - 1
```

Binary Search Algorithm

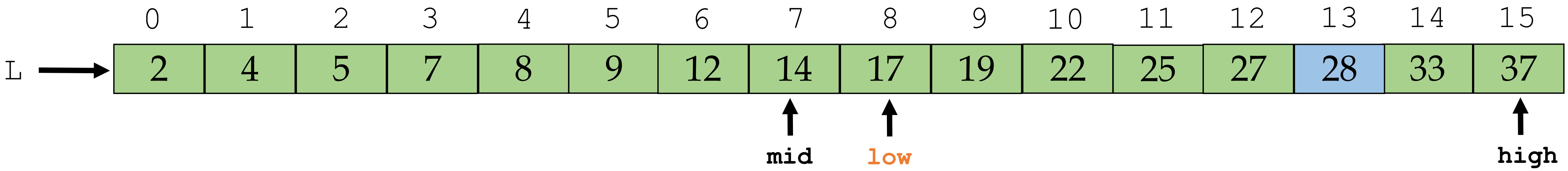


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L →	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{\text{low+high}}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
        Return mid
   Else If the value at the mid position is less than the target value
        Set low = mid + 1
   Else If the value at the mid position is greater than the target value
        Set high = mid - 1
```

Binary Search Algorithm

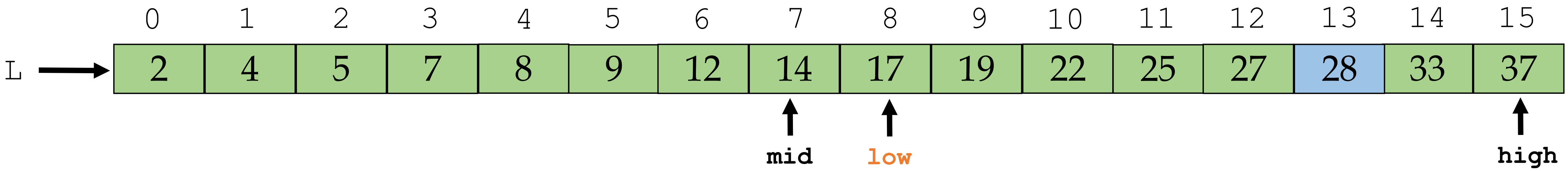


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{\text{low+high}}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
           Return mid
   Else If the value at the mid position is less than the target value
           Set low = mid + 1
   Else If the value at the mid position is greater than the target value
           Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above
```

Binary Search Algorithm



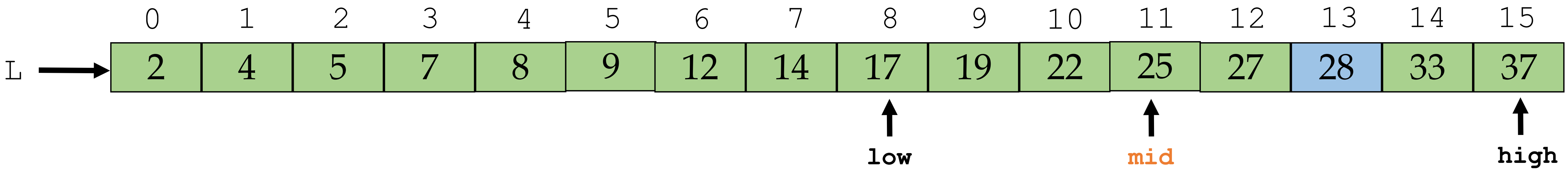
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L →	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37
								mid	low							high

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{\text{low+high}}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
           Return mid
   Else If the value at the mid position is less than the target value
           Set low = mid + 1
   Else If the value at the mid position is greater than the target value
           Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above
```

In Python this means, `while low <= high:`

Binary Search Algorithm

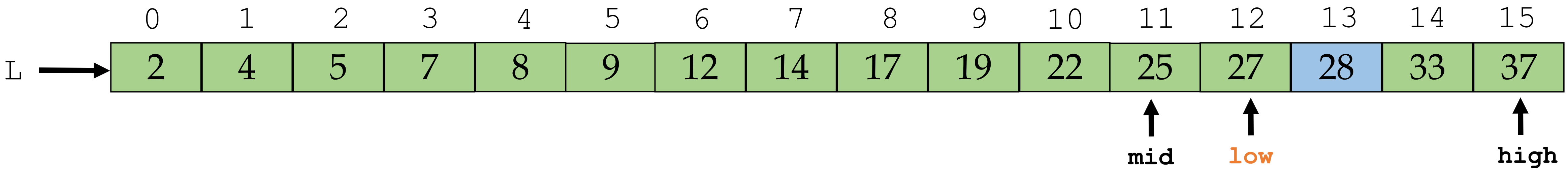


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Pseudo-code: **(target value is 28)**

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{\text{low+high}}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
           Return mid
   Else If the value at the mid position is less than the target value
           Set low = mid + 1
   Else If the value at the mid position is greater than the target value
           Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above
```

Binary Search Algorithm

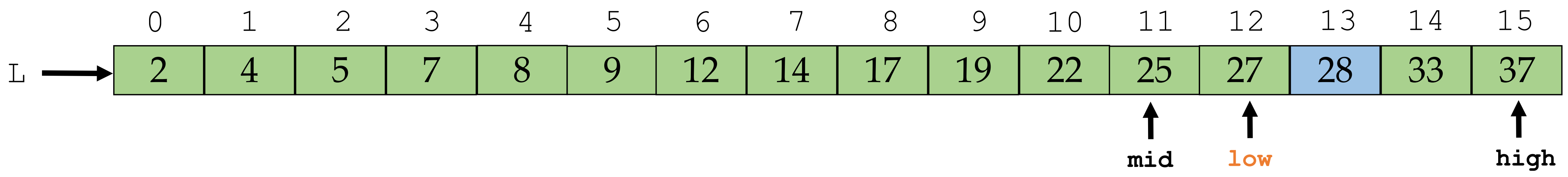


L	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37
												mid	low			high

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{\text{low+high}}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
          Return mid
   Else If the value at the mid position is less than the target value
          Set low = mid + 1
   Else If the value at the mid position is greater than the target value
          Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above
```

Binary Search Algorithm

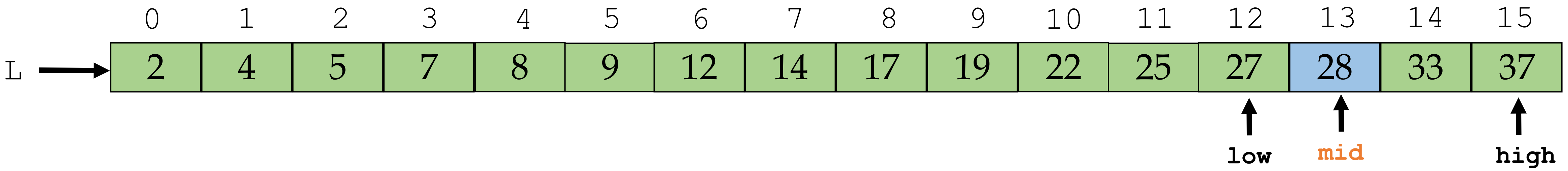


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37
												mid	low			high

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
        Return mid
   Else If the value at the mid position is less than the target value
        Set low = mid + 1
   Else If the value at the mid position is greater than the target value
        Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above
```

Binary Search Algorithm

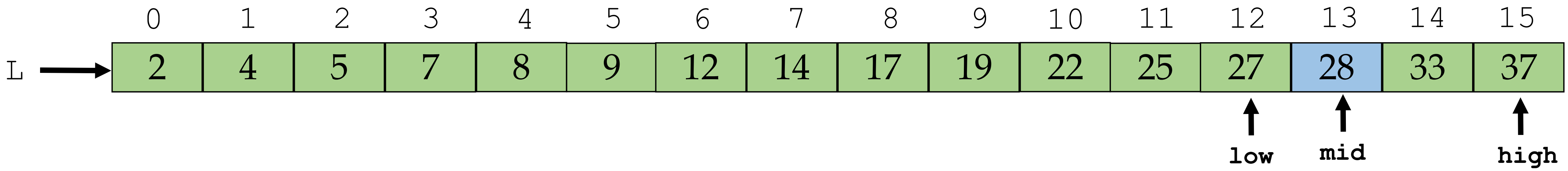


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
            Return mid
   Else If the value at the mid position is less than the target value
            Set low = mid + 1
   Else If the value at the mid position is greater than the target value
            Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above
```

Binary Search Algorithm

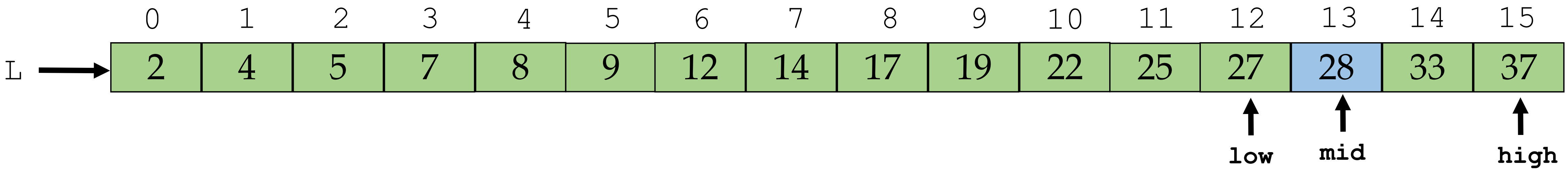


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
            Return mid
   Else If the value at the mid position is less than the target value
            Set low = mid + 1
   Else If the value at the mid position is greater than the target value
            Set high = mid - 1
5. As long as low doesn’t ‘cross over’ high, go back to step 3 above
```

Binary Search Algorithm

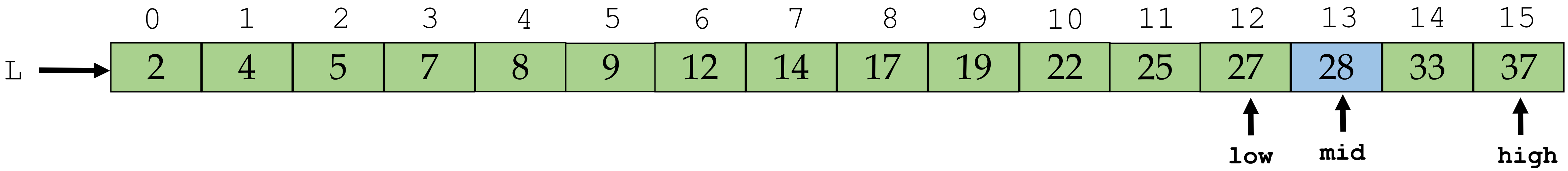


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{\text{low+high}}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
           Return mid
     Else If the value at the mid position is less than the target value
           Set low = mid + 1
     Else If the value at the mid position is greater than the target value
           Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above
```

Binary Search Algorithm



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L →	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37
													low	mid		high

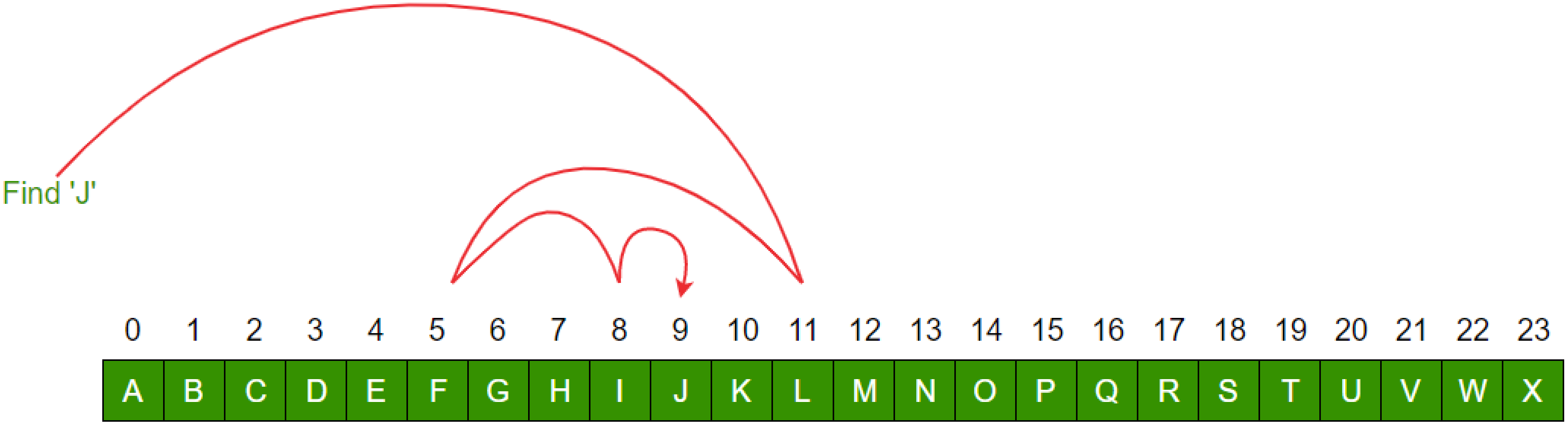
Pseudo-code: (target value is 28)

```
1. Set low = 0
2. Set high = length of list - 1
3. Set mid = $\frac{\text{low+high}}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
           Return mid
     Else If the value at the mid position is less than the target value
           Set low = mid + 1
     Else If the value at the mid position is greater than the target value
           Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above
```

13 is returned (as it is the value of mid)
This is the index of the target element.

Q. How many comparisons were needed?

Another Example



Find 'J'

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X

Tea/Coffee

An Roinn Oideachais
Department of Education