Computer Science Unplugged

An enrichment and extension programme for primary-aged children

Created by
Tim Bell, Ian H. Witten and Mike Fellows

Adapted for classroom use by
Robyn Adams and Jane McKenzie

Illustrated by Matt Powell

Parents’ Edition by Andrea Arpaci-Dusseau

Solutions
Solutions

Activity 1: Binary Numbers ................................................................. 1
Activity 2: Representing Bits ............................................................. 3
Activity 3: Representing Letters ......................................................... 4
Activity 4: Representing Pictures ....................................................... 5
Activity 5: Compressing Text ............................................................. 7
Activity 6: Detecting Errors ............................................................... 9
Activity 7: Detecting More Errors ...................................................... 10
Activity 8: Information Theory ......................................................... 11
Activity 9: Searching .................................................................. 12
Activity 10: Sorting .................................................................. 13
Activity 11: Faster Sorting ............................................................... 14
Activity 12: Sorting Networks ........................................................... 15
Activity 13: The Muddy City ............................................................. 17
Activity 14: Routing and Deadlock .................................................... 19
Activity 15: Finite-State Automata .................................................... 20
Activity 16: FSA Representation ....................................................... 22
Activity 17: Programming Languages .............................................. 24
Activity 1: Binary Numbers

a) Answers will vary. Possibilities include playing games, browsing the web, buying things on-line, writing stories, drawing pictures, listening to music, and watching movies.

b) Each card to the left has twice as many dots as the previous card.

c) 16 times 2, which is 32.

d) 32 times 2, which is 64.

e) We are multiplying the number of dots by 2.

f) 1: Only the first card with 1 dot on it should be face up.

g) 6: The card with 4 dots and the card with 2 dots should be face up.

h) 15: The cards with 8, 4, 2, and 1 should all be face up.

i) 21: The cards with 16, 4, and 1 dots should be face up.

j) 30: The cards with 16, 8, 4, and 2 dots should be face up.

k) No, there is only one way to form each number.

l) The biggest number is 16+8+4+2+1 which is 31. The other way to arrive at the answer 31 is to see that the next card would have 32 dots.

m) The smallest number is 0. (Not 1!)

n) No, all numbers between 0 and 31 can be made.

o) The general way to count is that the “least significant” card (that is, the card with the fewest dots) will be flipped for each number; whenever all lower cards are face up, the next card will be flipped up, and the lower cards will be flipped down.

p) 5 bits.

q) Because 8 + 1 = 9.

r) 01101: The card with 16 dots is face down, the cards with 8 and 4 dots are face up, the card with 2 dots is face down, and the card with 1 dot is face up.

s) 01101 = 8 + 4 + 1 = 13.

t) 00110 = 4 +2 = 6.

u) 01110 = 8 + 4 + 2 = 14.
v) \(10001 = 16 + 1 = 17\).
w) \(1 + 2 = 3\).
x) \(1 + 2 + 4 = 7\).
y) \(1 + 2 + 4 + 8 = 15\)
z) \(1 + 2 + 4 + 8 + 16 = 31\).

aa) The sum of the previous cards is always equal to the next higher card.
bb) 010, or 2, changes to 0100, or 4.
cc) 101, or 5, changes to 1010, or 10 in decimal.
dd) 110, or 6, changes to 1100, or 12 in decimal.

ee) A binary number is multiplied by two when a zero is shifted into the right-most digit.

**Activity 1 Worksheet 1**

What is **00110** (as a shortcut, this can be written **110**)? 6

What is **01100** (as a shortcut, this can be written **1100**)? 12

What is **10101**? 21

What is **11111**? 31

What is 8? 01000

What is 3? 00011

What is 17? 10001

What is 30? 11110

What is 14? 01110
Activity 2: Representing Bits

a) Answers will vary. Anything with two different states can represent a bit. Examples include: whether a window is open or closed, whether a flag is up or down, whether a beep is loud or quiet, or whether a fork is pointed up or down.

b) It may be useful to check using the cards from the previous exercise, but this shouldn't be necessary.

c) The magnetic disk is recording binary number 0010110. (This is equal to 16+4+2=22 in decimal.)

d) The optical disk is recording 0101101101001010 in binary.

e) 0111 will be a low pitched beep, followed by three high beeps.

f) 0101 will be a low, high, low, high sequence of beeps.

g) 2 is equal to 0010, which is two low beeps, a high beep, and a low beep.

h) 9 is 1001, which is a high beep, two low beeps, and a high beep.

Activity 2 Worksheet 1: Working With Binary

01001 = 9
101 = 5
00000 = 0
10 = 2
1 = 1
1010 = 10
1101 = 13
10001 = 17
10100 = 20
11111 = 31
Activity 3: Representing Letters

a) The letter d is the decimal number 4.
b) The letter y is 25.
c) 19 represents s.
d) 5 represents e.
e) “hello” is 8 5 12 12 15
f) “1 14 14 1” is “A N N A”
g) Answers will vary.

Activity 3 Worksheet 1: Decoding Messages
Coded message: HELP IM TRAPPED
Activity 4: Representing Pictures

a) Examples will vary. Possibilities include storing photographs from cameras, scenes from movies, cartoons in games, pictures on web pages.

b) 3rd line: 0, 1, 1, 1, 1

c) 5th line: 1, 0, 0, 0, 1

d) Last line: 0, 1, 1, 1, 1

e) 3rd line: 1, 4

f) 5th line: 0, 1, 3, 1

g) Last line: 1, 4

h) You will need to use two numbers that add up to 10, with 0 white pixels in between. For example: 7, 0, 3.
Activity 4 Worksheet 1: Decoding Pictures

[Grid and images of various pixelated pictures: a cup, a planet, and a face]
Activity 5: Compressing Text

a) First line: “tter” is repeated.
   Second: Whole line is repeated.
   Third: “the rain” repeated from title
   Fourth: Whole line
   Fifth: Whole line
   Sixth: “the”

b) The poem should look as follows.

```
The Rain
Pitter pa
Listen to
On window pane
```

c) Original poem first line: 8
   Second: 13
   Third: 13
   Fourth: 18
   Fifth: 13
   Sixth: 13
   Seventh: 18
   Total: 96 letters and spaces

d) Compressed poem first line: 8
   Second: 9
   Third: 0
   Fourth: 10
   Fifth: 0
   Sixth: 0
Seventh: 14
Total: 41

e) The compressed poem is $\frac{41}{96}$ the size of the original, which is a little less than half the size. Or, you could say it has $96 - 41 = 55$ fewer letters and spaces.

f) 5 arrows

g) $5 \text{ arrows} \times 2 \text{ characters/arrow} = 10 \text{ characters}$

h) $41 + 10 = 51$.

i)

j) The compressed poem is $\frac{51}{96}$ the size of the original, which is a little more than half the size. Or, you could say it has $96 - 51 = 45$ fewer letters and spaces.

Activity 5 Worksheet 1: Say that again!

*Pease porridge hot,*
*Pease porridge cold,*
*Pease porridge in the pot,*
*Nine days old.*

*Some like it hot,*
*Some like it cold,*
*Some like it in the pot,*
*Nine days old.*
Activity 6: Detecting Errors

a) The extra cards are called parity cards; they are set up so that each row and column has an even number of face-up cards. For example, the first row parity card is set face down so that there will remain two face up cards; the second row parity card is set face up so there will be size face up cards. The first column parity card is face down so that there will remain two face up cards; the second column parity card is set face up so that there will be four face up cards.

b) You can find the flipped card by looking for the row with an odd number of face up cards and then the column with an odd number of face up cards; the card in that row and column is the one that must have been flipped.

c) You can find the flipped card by looking for the row with an odd number of face up cards and then the column with an odd number of face up cards; the card in that row and column is the one that must have been flipped.

d) Student should be able to identify a single flipped data card.

e) The computer should fix the problem by flipping the bit back before it gives the data to the user.

f) Student should be able to identify a single flipped parity card.

g) If a parity bit is flipped, there isn’t a problem with the data that the user cares about. In the case of sending messages, it is generally safe to just ignore the fact that a parity bit was flipped. No, there was not a problem with

h) Student should be able to identify that no card was flipped.

i) Set up a random 5 x 5 array. Check that the child computes the correct parity cards.

j) Take turns while child is interested.)
Activity 7: Detecting More Errors

a) Answers will vary. Check that the parity cards are correct.

b) The student will not be able to determine which two cards have been flipped, but they will be able to narrow it down to four possibilities.

c) Again the student will not be able to tell which two cards have been flipped, but they will be able to determine the two bad columns. They should be able to infer that the two cards must be in the same row.

d) When the computer detects an error it cannot correct, it should ask the sending computer to resend the message.

e) Answers will vary. Work through the logic to determine where the bit flips could have occurred.

f) The student can detect that there are 4 bad columns, but cannot see which row (or 2 rows with 2 errors each) have the errors.

g) The student will not be able to detect any errors. All of the parity values remain correct!

h) Yes, the last digit is 4.

i) 1-00-235045 → 1x10 + 0x9 + 0x8 + 2x7 + 3x6 + 5x5 + 0x4 + 4x3 + 5x2 = 89
   89 / 11 = 8 with a remainder of 1. 11-1 = 10, which has a checksum character of X.
Activity 8: Information Theory

a) Answers will vary. Information is anything that is useful or interesting to know. Possible examples include the contents of an encyclopedia, dictionary, phone book, or atlas; what someone had for dinner last night, how much something costs, directions for getting to school, the rules for playing a game.

b) Answers will vary. Possibilities include counting the number of pages, sentences, or words in the book. Other possibilities could be how many things you found interesting or surprising in the book, or how long you could read the book before you wanted to put it down.

c) From most information to least:
   • 1000 pages of a telephone book
   • 500 pages of a telephone book
   • 1000 pages of the words “blah, blah, blah” repeated over and over
   • 1000 pages of blank paper

d) Answers will vary. Target answer is by how hard it is to guess the message.


f) Unless the student is performing a “binary search”, he or she will probably need more than 7 guesses sometimes.

g) Use the algorithm described in the text.

h) Answers will vary.

i) Answers will vary.

Activity 8 Worksheet 1: Decision Trees

1. What are the yes/no decisions, or answers, needed to ‘guess’ the number 5?
   Yes, No, Yes

2. How many yes/no decisions do you need to guess any number between 0 and 7? Three; one decision at each level of the tree.

3. The bits to represent each number exactly match the decisions to get to that number in the tree! number in binary. Look closely at the tree. If no=0 and yes=1, what do you see?
Activity 9: Searching

a) Answers will vary.

b) Answers will vary. Best possible score is 1 guess.

c) Worst possible score is 25 guesses.

d) Requires 25 guesses if secret ship is not there.

e) Usually need to look at about half the ships, or about 12 guesses.

f) About half the ships, or 50 guesses.

g) Answers will vary.

h) Answers will vary. Best possible score is 1 guess.

i) The worst possible, if using the binary search correctly on 25 ships, is 5 guesses.

j) Still takes 5 guesses.

k) It will usually take about half of the possible ships, or 2 or 3 guesses.

l) Figure out the bucket of the student’s secret ship out loud. Have the student figure out in which bucket your secret ship is located.

m) Answers will vary.

n) Best possible is 1 guess.

o) Worst possible is the number of ships in a given bucket. If all of the ships were in the same bucket, the game becomes the same as the linear search with completely secret ships.

p) Would have to search through all of the ships in the bucket.

q) With 25 ships and 10 buckets, there should be about 2.5 ships per bucket. We should then find the ship in 1 or 2 guesses.

r) Pick the bucket with the most ships in it.

s) The bucket with the fewest ships.

l) The only advantage of linear searching is that it doesn’t require that the ships are organized ahead of time in any way. Hashed searching is generally the fastest, as long as one has enough buckets for all of the ships; if one is worried about the case where too many ships happen to fall into the same bucket (in which case the search would be very slow), then binary searching would be better.
Activity 10: Sorting

a) Sorting with letters is important for dictionaries and phone books; if not sorted, would be very difficult to lookup a word or a name. Sorting with numbers is also important: for example, looking at all of the grades that students received and seeing who did well and who needs more help. Even web search engines sort results so that you can see the pages that best match your search terms at the top of the list.

b) The easiest way is to compare two objects, keeping the lightest and putting the other aside. Continue to compare each object to the lightest so far, until you’ve seen all of them.

c) The number of comparisons is equal to the number of objects minus 1, or 7.

d) Compare two objects and put them in order. Then, compare the 3rd object to one of them (e.g., the lightest). If the 3rd object is lighter than the lightest, we are done. If the 3rd object is heavier, then compare it to the heavier object and place it accordingly.

e) If you get lucky, you will need only two comparisons, but you might need 3.

f) Selection sort: To find the minimum of two objects you need one comparison, three needs two, four needs three, and so on. To sort eight objects using selection sort takes 7 comparisons to find the first one, six to find the next, five to find the next and so on. That gives us: 7 + 6 + 5 + 4 + 3 + 2 + 1 = 28 comparisons.

g) Insertion sort: Answers will vary. In the best case (though this is very unlikely), you will only need 7 comparisons (e.g., if you always happen to pick the lightest remaining object in the unsorted pile and you begin by comparing with the lightest sorted object). In the worst case, you will need 28 comparisons (e.g., you always happen to pick the heaviest remaining object and you always begin comparing with the lightest objects); this is the same as selection sort. In typical cases, you will require a number of comparisons between the two.

h) Bubble sort: Again, answers can vary. And again, in the best case, it is possible that only 7 comparisons will be needed, though this only occurs when the list is already sorted before you begin the sorting algorithm! Typical cases are likely to require about 28 comparisons (again); 7 comparisons are required to “bubble” the heaviest weight to its proper position; 6 to bubble the next heaviest weight up, and so forth.

i) The fastest of these three algorithms will be insertion sort in most cases.

j) For these algorithms, given twenty objects, the worst case number of comparisons will be 19+18+17+..+1 = 190 comparisons.
Activity 11: Faster Sorting

a) Quick sort: Answers will vary. If you are lucky, then quicksort will require only about 14 comparisons. If you are unlucky, then you may need up to the full 28 comparisons.

b) With quicksort, you will perform the fewest comparisons if you pick a pivot object that is in the middle of the rest of the objects; that is, the pivot object that divides the remaining objects into two piles of about the same size.

c) Merge sort will also require about 14 comparisons; even if you are unlucky, it should require only 17 comparisons.

d) Quicksort and mergesort are generally much faster (that is, require fewer comparisons) than selection, insertion, or bubble sort.
Activity 12: Sorting Networks

a) A: A sends its higher number to A
b) B: A sends its lower number to B
c) B: C sends its higher number to B
d) C: C sends its lower number to C
e) A: In step 2, A sends higher number to A
f) B: In step 2, A sends its lower number to B
g) Point to 3rd column of three nodes.
h) Point to 4th column, which contains only two nodes, A and B
i) 5th column, which contains only one node, B
j) In step 1, Node a: Sends 1 to A and 5 to B; Node b: Sends 6 to A and 3 to C; Node C: Sends 4 to B and 2 to C.
k) In step 2, Node A sends 6 to A and 5 to B; Node b send 4 to A and 1 to C; Node C sends 3 to B and 2 to C.
l) In step 3, A sends 6 to output, 4 to A; Node b sends 5 to A and 3 to B; Nodes C to 2 to B and 1 to output.
m) After 3 time steps, we know that the highest number is 6 and the lowest number is 1.
n) In step 4, node A sends 5 to output and 4 to B and B sends 3 to B and 2 to output.
o) In step 5, node B sends 4 and 3 to output.
p) Answers will vary.
q) The numbers will be sorted from lowest to highest (instead of highest to lowest).
r) The following network sorts three numbers.
s) The second one is faster. Whereas the first network requires all comparisons to be done serially, one after the other, the second has some being performed at the same time. The first network is an example of serial processing, whereas the second uses parallel processing to run faster.

Networks can also be used to perform other computations across a set of inputs.

t) The single output is the maximum of all of the inputs.

u) Yes, each new person and shovel can dig a new part of the ditch.

v) No, to dig a deeper ditch, one must wait for the earlier parts of the hole to be dug.

w) You can accelerate meal by having different people make different dishes or cut up different ingredients; you can't parallelize actual baking or cooking of a single dish, since it takes that long for it to bake.

x) Multiple pieces of clothing are all washed together. You can dry one load of laundry while another is being washed. But, you can't wash and dry the same piece of clothing at the same time!
Activity 13: The Muddy City

a) The path (1 tile) between house A and house B should be paved, plus the path (2 tiles) between house B and house C. This requires a total of three stones.

b) To walk from A to C on paved roads, one must walk from A to B to C.

c) This is one possible solution that paves 7 stones; other paths of 7 stones are possible.

d) To walk from A to D, one first goes to house B.

e) The following paths are also just as good, since all houses are still connected using 7 tiles.

f) Would prefer the direct path from B to E is paved, as in the first solution.

g) Planning how water pipes should be laid out in a city, or gas lines, or electrical lines.
h) 10

i) How the graph looks isn’t important, as long as house A is connected to B with a length of 1, house B to house C with a length of 2, and house C to A with a length of 3, and no other connections.

j) The graph is as follows.

![Graph Image]

**Activity 13 Worksheet 1: The Muddy City**

One good strategy to find the best solution is to start with an empty map, and gradually add tiles until all of the houses are linked, adding the paths in increasing order of length, but not linking houses that are already linked. Different solutions are found if you change the order in which paths of the same length are added. One possible solution is shown below.

![Diagram Image]

The following is the graph corresponding to this city

![Graph Image]
Activity 14: Routing and Deadlock

a) A total of 3 moves are required, as follows:
   1: B sends msg A to A.
   2: A sends msg B to B
   3: B sends msg A to A.

b) No, if each computer starts with two messages, then there is no empty space in
   which to hold a new message. Does the game work if there are two messages
   destined for both A and B? Why or why not?

c) Answers will vary. If you get stuck, you may find the puzzle easier if you imagine
   that there are no network connections between computers A and C.

d) Answers will vary. If you get stuck, you may find the puzzle easier if you imagine
   that there are no network connections between computers A and D.

e) No.

f) No, there is no longer a path between computer B and its messages; there is no
   available space in computers A or C and they will not let go of any of their
   messages.

g) Yes, messages can still be routed between B and D through A if only C is greedy.

h) Deadlock happens in the real world. For example, imagine two people in a fight,
   where each says to themselves “I’ll say I’m sorry after she does”. Neither one
   will ever apologize and they will never make up! As another example, imagine
   when four cars arrive simultaneously at an intersection with stop signs: if each
   waits for the car on its right to go first, no car will ever be able to go!
Activity 15: Finite-State Automata

a) The following is a map between the three islands:

![Map of islands](image)

b) Answers will vary. One possible path is from Pirate’s Island take ship B to Musket Hill, then ship B to Mutineers’ Island, then ship A to Smugglers’ Cove, and finally ship B to Treasure Island.

c) The following is a complete map.

![Complete map](image)

d) The quickest route from Pirate’s Island is take ship B to Musket Hill, then ship B to Mutineers’ Island, then ship A to Smugglers’ Cove, and finally ship B to Treasure Island.
e) There are many possible slow routes. One slower route that hits all of the islands would be from Pirates’ Island take ship A to Shipwreck Bay, ship B to Dead Man’s Island, ship A to Musket Hill, ship B to Mutineers’ Island, ship A to Smuggler’s Cover, and B to Treasure Island.

f) Lots of paths involve loops. For example, one could travel between Pirates’ Island and Musket Hill many times before proceeding to Mutineers’ Island.

g) Answers will vary.

h) Answers will vary.

i) Answers will vary.
Activity 16: FSA Representation

a) The most efficient path can be denoted simply BBAB.

b) An example loop is BABABABABBAB

c) Yes.

d) Yes.

e) No.

f) AB: Yes.

g) BABAA: Yes

h) ABBBA: No

i) AAABABA: Yes

j) AAABA: No

k) Strings containing an odd number of A’s are accepted, since A moves one in and out of the accept state, whereas B’s do nothing.

l) AB: Yes

m) BA: No

n) ABAB: Yes

o) AABB: No

p) ABABA: No

q) ABBA: No

r) Only strings that completely and exclusively repeat the pair ”AB” some number of times are accepted.

s) B: Yes

t) BBB: Yes

u) BBA: Yes

v) ABBA: Yes

w) AAA: No

x) Any string with at least one B is accepted.
y) An example sentence is: a friendly dog and the old pirate laughed.

z) Answers will vary.
Activity 17: Programming Languages

a) No, this would not be good!

b) When giving directions to someone, often leave out many details. Like “go through that door” and don’t first say “open the door”.

c) The picture should look something like this:

![Diagram of a person's name at the center of an X]

Your name

d) Hopefully the pictures look fairly similar.

e) When the person describing the picture doesn’t get to look at what is being drawn, mistakes can’t be fixed. There is no way for the describing person to know that there is a problem if they can’t see the picture.

f) This is the hardest case because there is only one-way communication between the describer and the drawer; if there are any problems, the describer won’t know until the very end.

g) It is important that programs are well written. A small error can cause a lot of problems. With an error, the space shuttle could miss its destination, the power plant could blow up, or two trains could hit each other. In all cases, people could die!