Discrete Mathematics
Practical activities for Algorithms

Aim	To develop students' understanding of why algorithms are necessary when problems become larger and computers are involved in the solution	
Resources required	A projector and computer able to show the Bin-packing power-points from Integral Maths Packs of cards with the aces and picture cards removed Bin-packing activity materials; problem, grids & cards (or wooden pieces and grids) Sorting algorithms explanation sheet	
Activity		**Key questions**
Teacher presentation	The students are shown the First-fit Bin-packing algorithm using the Integral Maths power-point with the teacher narrating. Students are asked to suggest a better approach. Students are shown the First-fit decreasing Bin-packing algorithm as an alternative approach and encouraged to reflect on which method is more efficient.	Is this a reasonable approach to take? When might this approach be less successful? Is First-fit decreasing a 'better' method? Why?
Student pair task	Students work in pairs on two versions of the same task, without being aware that they are solving the same problem. Pairs in group one have a set of 12 rods of different lengths to fit into bins of size 12. These are drawn one at a time from an opaque bag or envelope. They use First-fit bin-packing. Pairs in group two have their rods in a clear bag and can start immediately with the complete set. They will use First-fit decreasing Bin-packing. During the activity the teacher should move around the class asking key questions. For an extension to the problem, the teacher can ask if the solution found using these algorithms is optimal or not – does trial and improvement give a better solution?	Why is it appropriate sometimes to use First-fit rather than First-fit decreasing? Does First-fit decreasing always give a better solution? Does First-fit decreasing always give an optimal solution? What practical considerations would need to be taken into account if you had to deal with a large set of data?
Teacher presentation	The teacher demonstrates the Bubble Sort using 8 numbered cards from one suit. Only two cards should be	How many comparisons will there be in each pass?

	handled at any one time, mirroring computer comparisons. Students can be assigned roles of counting passes and swaps. The algorithm should be repeated until students are comfortable with how it works.	How will this change with each pass? How many passes will be needed at most for sorting 8 numbers? What if there were *n* cards?
Student activity	Students try out this algorithm for themselves in pairs; one performs the algorithm whilst the other checks it is being carried out correctly, then they exchange roles.	
Plenary and Teacher presentation (2)	The teacher then demonstrates the algorithm with a larger set of cards starting with them face down, only turning over the two to be compared at any one time. Once a card is sorted to the end of the row it should be left face up. This reinforces the understanding that a computer can only compare two numbers at a time and if the set of numbers increases in size, it is not easy to simply scan for the largest at a glance. This also reinforces the need for a stopping condition: a pass with no swaps. As an extension the students could think of their own variations of the algorithm or try one of the other sorting methods for comparison.	What if *n* were a really big number? How do we know when all the cards are sorted? Why do we need an algorithm at all for sorting numbers into order?

Teaching Discrete Mathematics

Activity: Bin Packing

Bin Packing

Where objects of varying sizes must be placed into containers of a fixed capacity, the problem is described as 'bin packing'. The name is used for any problem of this general type, whether to do with objects, lengths, times or whatever the scenario.

Two algorithms are commonly used to attempt to solve these problems:

First-Fit Bin Packing:

Number the bins, then always place the next item in the lowest numbered bin which can take that item.

First-Fit Decreasing Bin Packing:

Reorder the items into decreasing order of size. Number the bins, then always place the next item in the lowest numbered bin which can take that item.

Activity

A builder uses piping of standard length 12 metres.
The following sections of varying lengths are required for a particular job:

Section	A	B	C	D	E	F	G	H	I	J	K	L
Length (metres)	2	2	3	3	3	3	4	4	4	6	7	7

Explore how many standard 12 m lengths of pipe will be required if each of the following methods is used:

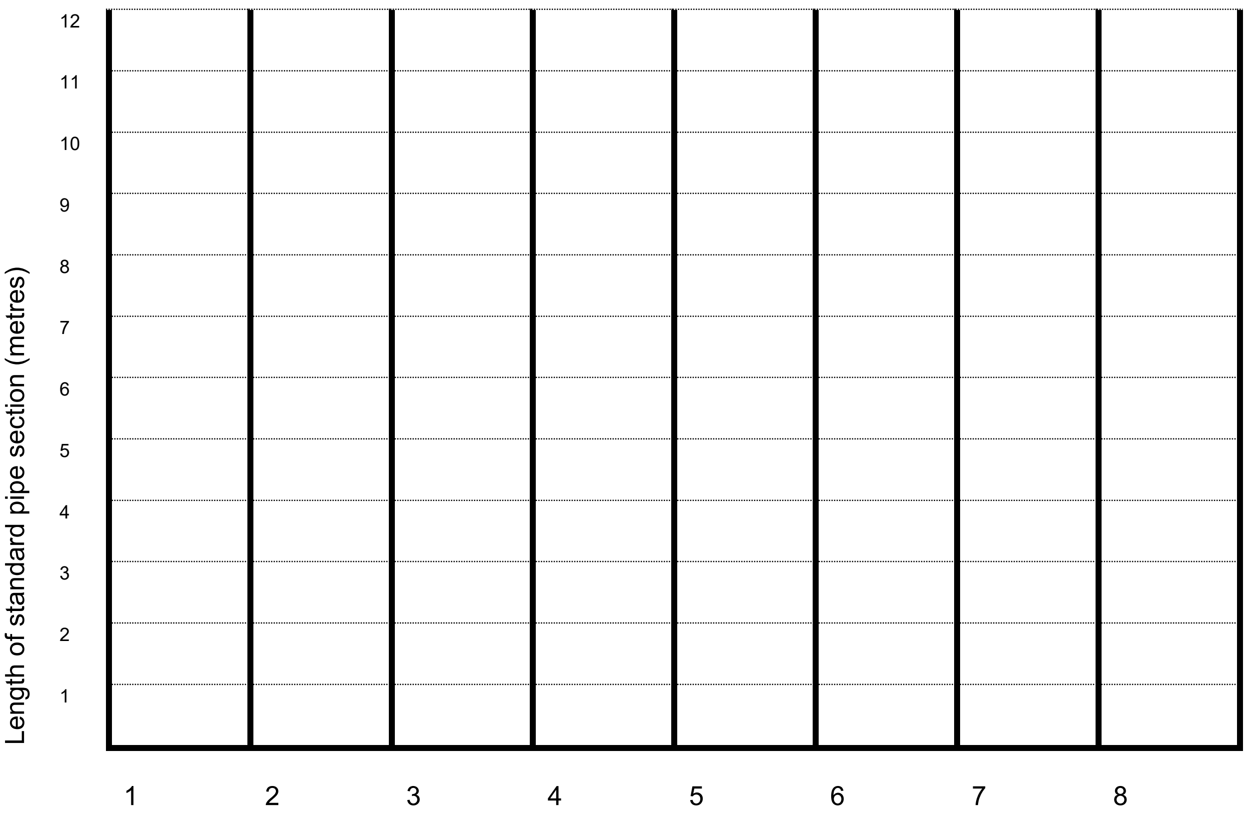
First-fit bin packing

First-fit decreasing bin packing

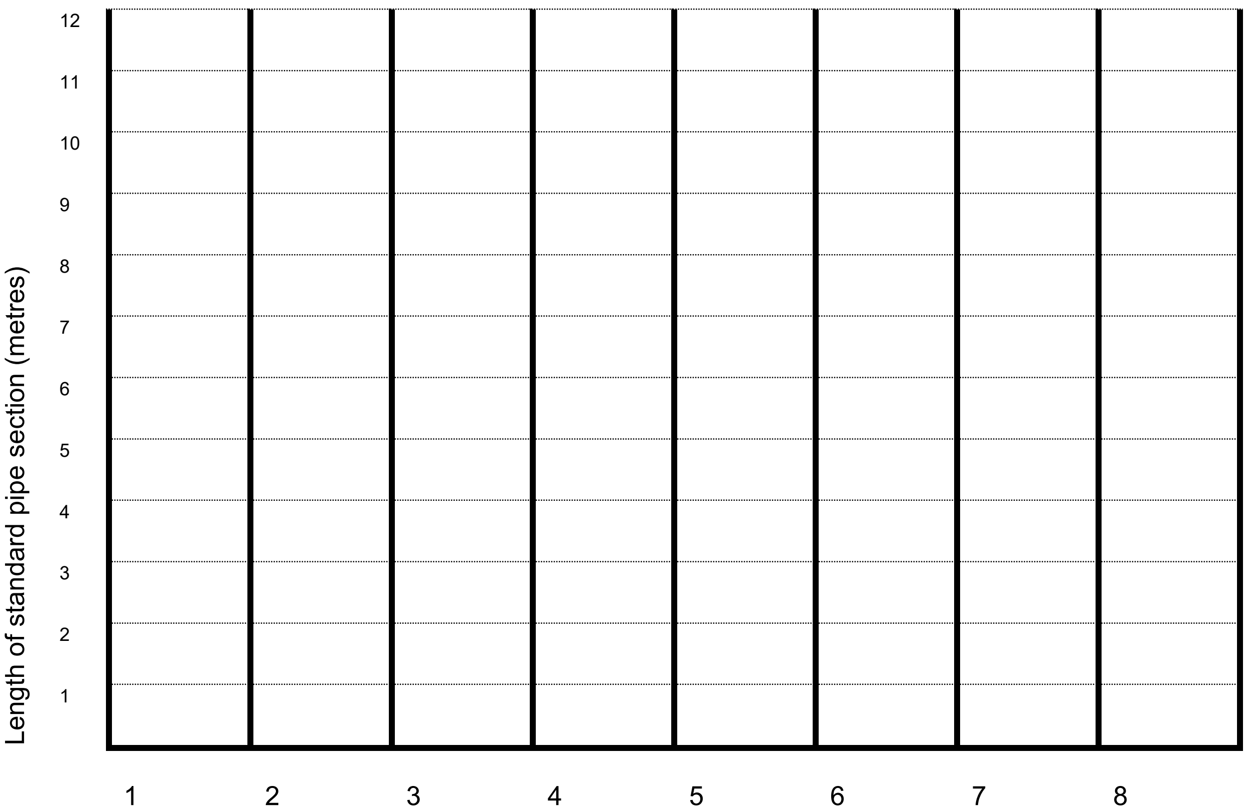
Trial and improvement

[Adapted from p. 209-210, AEB Discrete Mathematics, Heinemann, 1992]

Bins

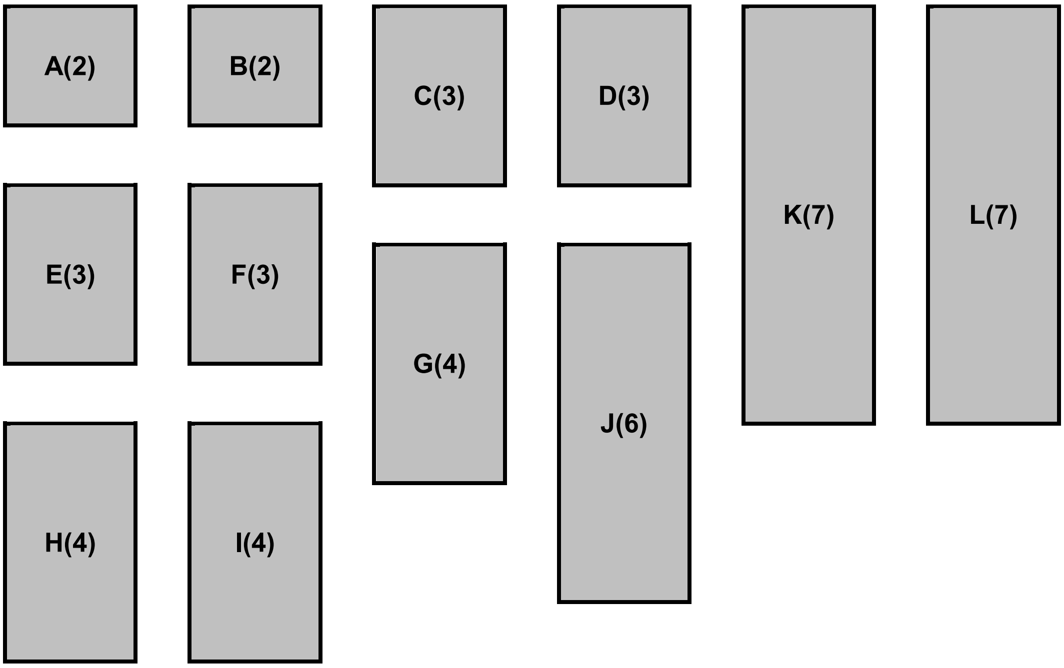


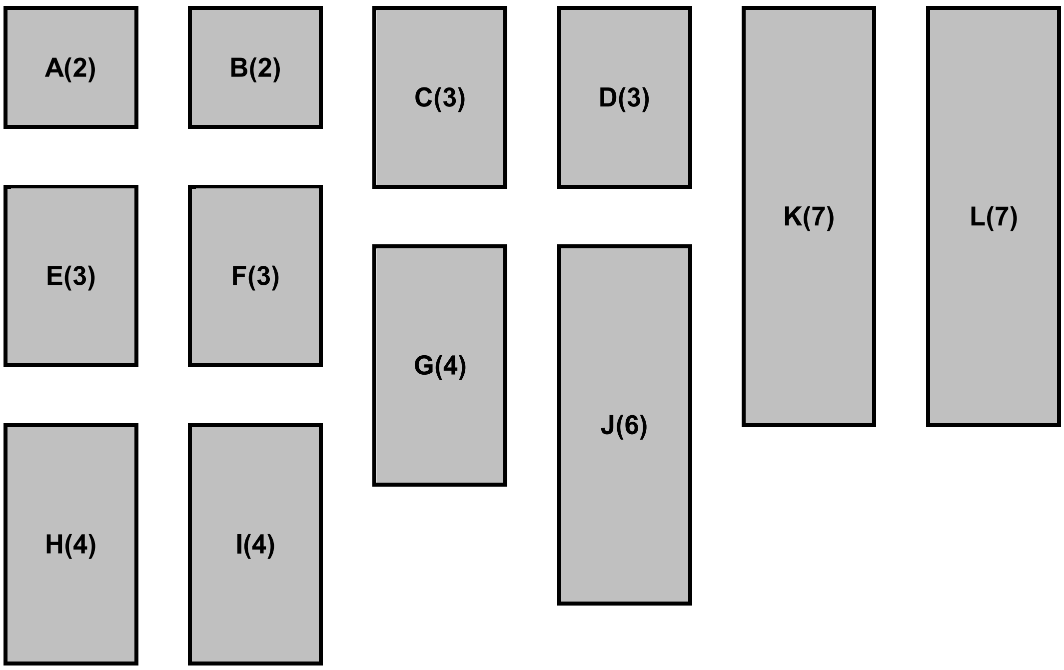
Length of standard pipe section (metres)
12
11
10
9
8
7
6
5
4
3
2
1
1
2
3
4
5
6
7
8



Length of standard pipe section (metres)
12
11
10
9
8
7
6
5
4
3
2
1
1
2
3
4
5
6
7
8

Pipe Sections

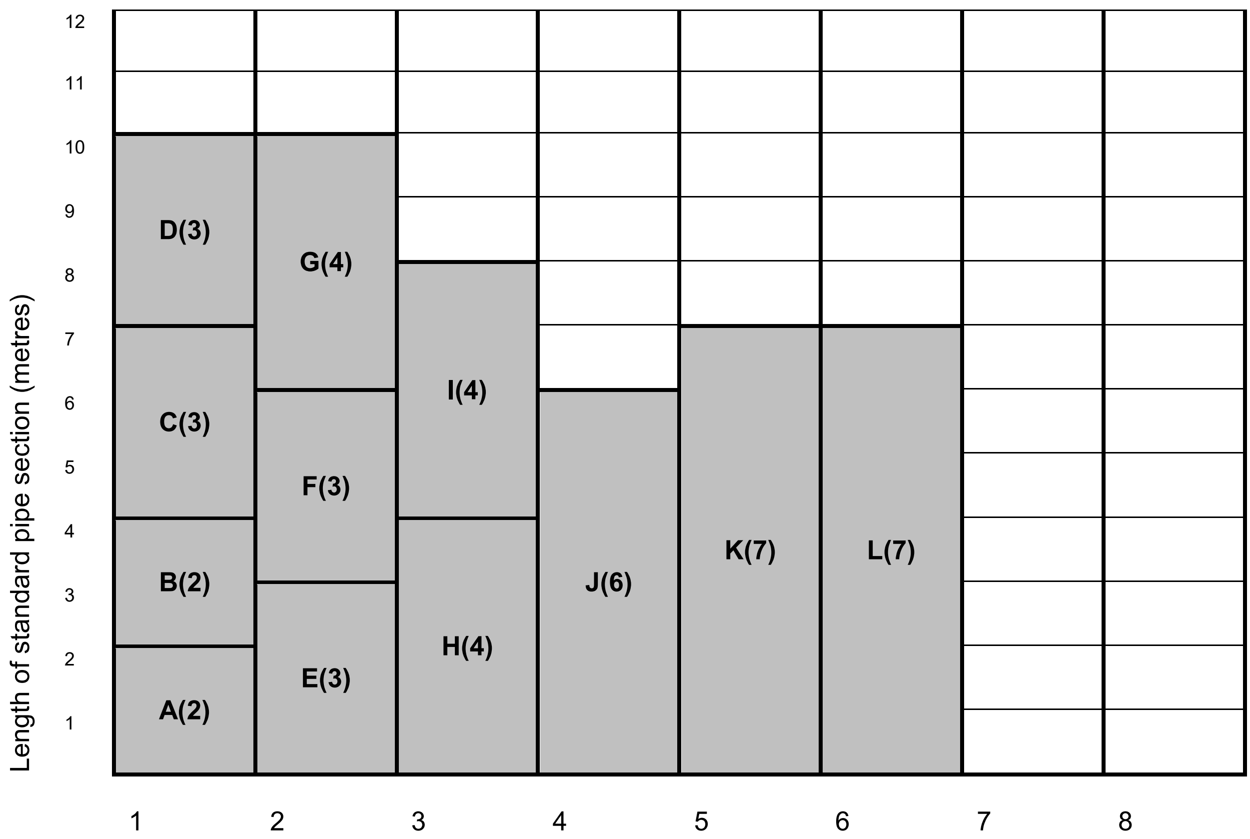




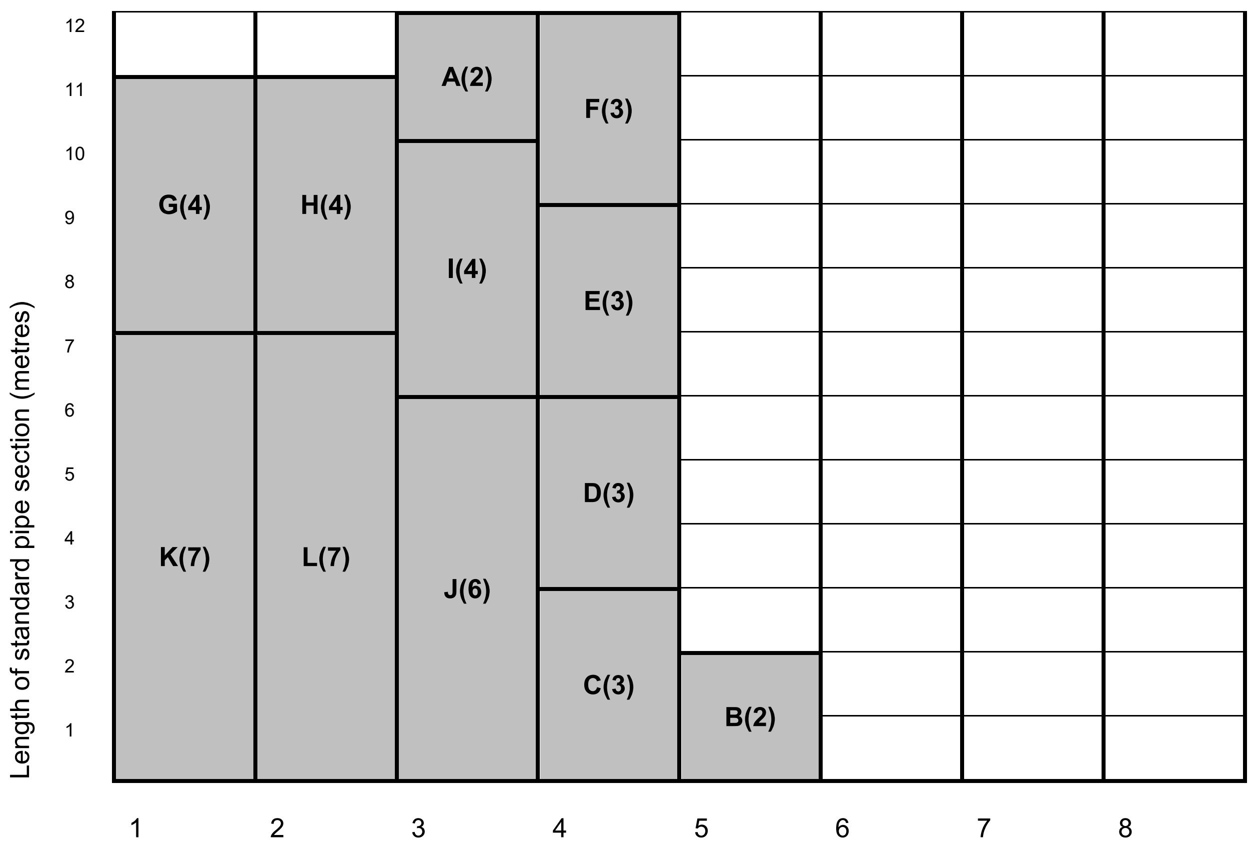
Cuisinaire rods can be used as a suitable alternative to represent the pipe sections. Grids scaled for this purpose are found at the end of this document.

Solutions

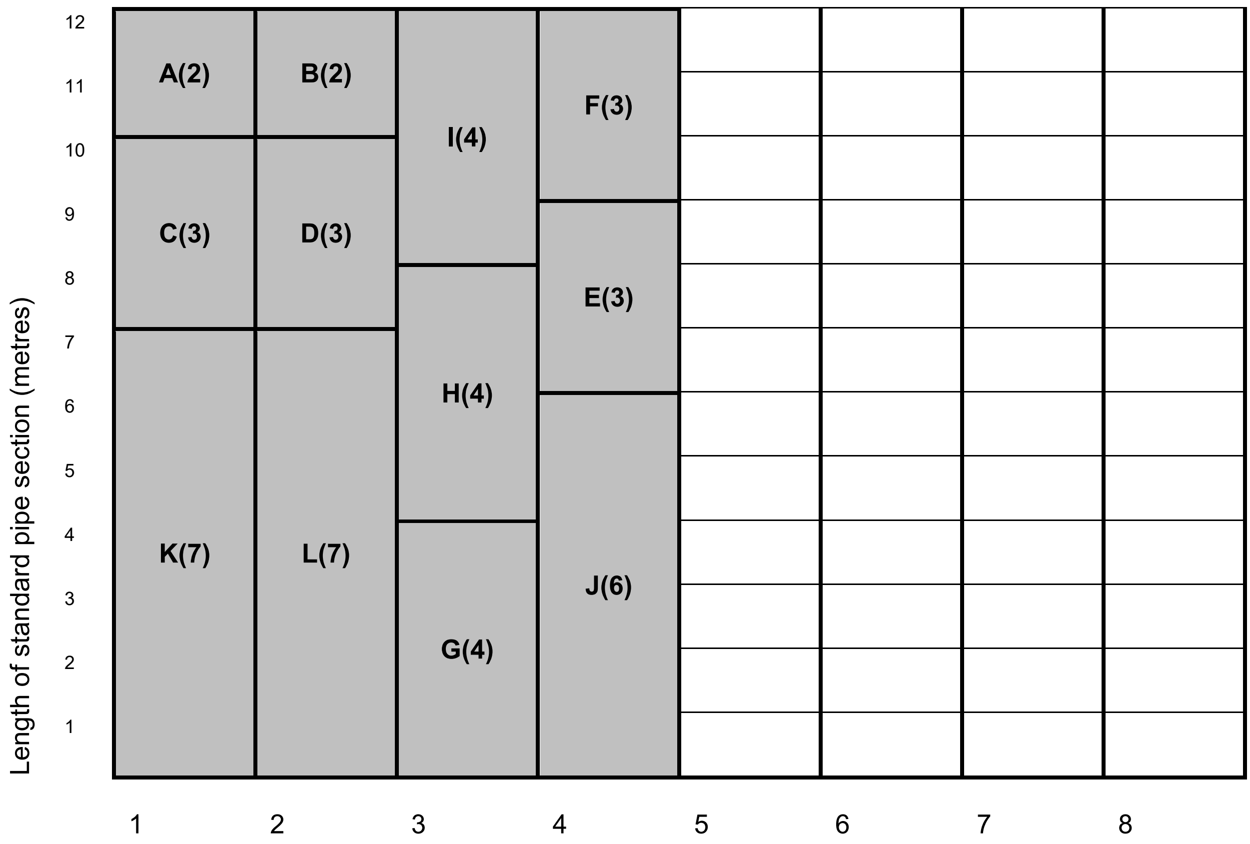
(a) First-fit bin packing



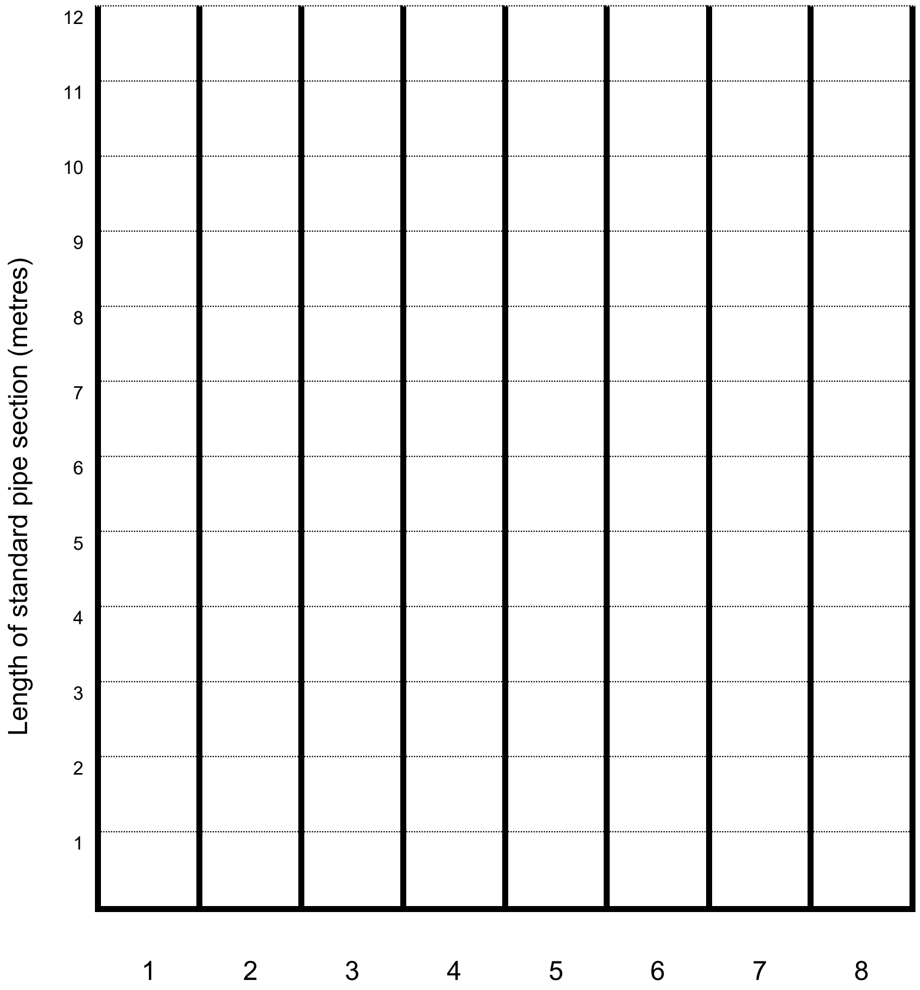
(b) First-fit decreasing bin packing

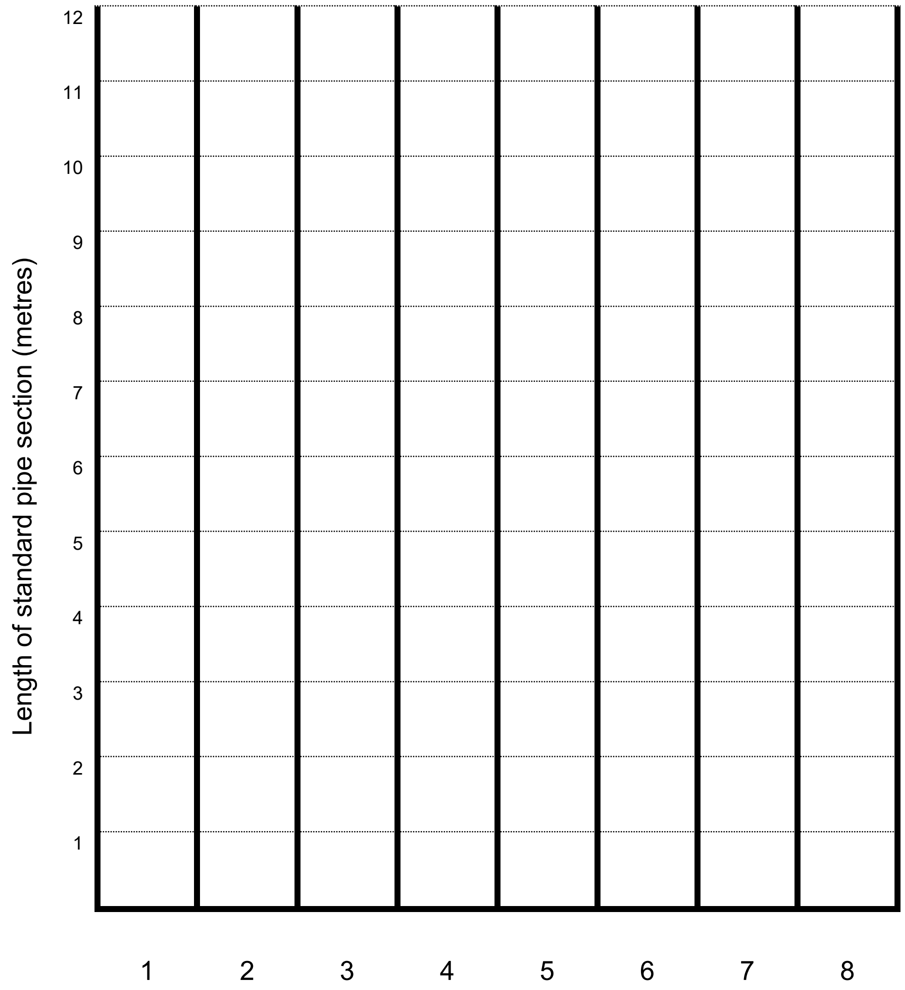


(c) Trial and improvement



Bins – scaled for use with cuisinaire rods





Teaching Discrete Mathematics

Algorithms for Sorting

Bubble Sort

The Bubble Sort works repeatedly from left to right, comparing pairs of numbers and exchanging so that the larger number is on the right.

Step 1: Compare the first two numbers.

Step 2: If the first number is larger than the second, exchange the numbers.

Step 3: Repeat steps 1 and 2 for all pairs of numbers until the end of the working list. The last number will be in its final position and is removed from the working list.

Step 4: Repeat steps 1 to 3 until no more exchanges are made.

Shuttle Sort

The Shuttle Sort works by comparing pairs of numbers and exchanging them if necessary.

Step 1: Compare the first two numbers and exchange if necessary.

Step 2: Compare the third and second numbers and exchange if necessary, then compare the second and first numbers and exchange if necessary.

Step 3: Compare the fourth and third numbers and exchange if necessary, compare the third and second numbers and exchange if necessary, compare the second and first numbers and exchange if necessary. The pass ends whenever a comparison is made but no exchange is required.

Step 4: For a list of length n, continue until n-1 passes have been performed.

Quick Sort

The Quick Sort works by taking a pivot from the list and arranging the numbers into sub-lists below and above the pivot. Each sub-list is then sorted around a new pivot from that sub-list.

Step 1: Choose any number x from the list L. This might be specified, sometimes as the first or last number but often as the middle number. When locating the middle of an even number of elements the convention is to select the right-hand of the middle pair, i.e. for N elements, x is the smallest integer such that $x \geq \frac{1}{2}(N + 1)$.

Step 2: Write all the numbers smaller than x to the left of x, reading the original list from left to right. These form a new list L1. Write all of the numbers larger than x to the right of x reading the original list. These numbers form a new list L2.

Step 3: Apply steps 1 and 2 to each separate list until all of the lists contain only one number. The original list is now in ascending order.

Try using each of these methods to sort: (a) 4, 7, 9, 6, 8, 5, 2, 3
and (b) 5, 8, 7, 2, 3, 9, 6, 4,

Solutions

In each case the list is shown after each pass

Bubble Sort

a) **4, 7, 9, 6, 8, 5, 2, 3**

After 1st pass:	4, 7, 6, 8, 5, 2, 3, 9
After 2nd pass:	4, 6, 7, 5, 2, 3, 8, 9
After 3rd pass:	4, 6, 5, 2, 3, 7, 8, 9
After 4th pass:	4, 5, 2, 3, 6, 7, 8, 9
After 5th pass:	4, 2, 3, 5, 6, 7, 8, 9
After 6th pass:	2, 3, 4, 5, 6, 7, 8, 9
After 7th pass:	2, 3, 4, 5, 6, 7, 8, 9

b) **5, 8, 7, 2, 3, 9, 6, 4**

After 1st pass:	5, 7, 2, 3, 8, 6, 4, 9
After 2nd pass:	5, 2, 3, 7, 6, 4, 8, 9
After 3rd pass:	2, 3, 5, 6, 4, 7, 8, 9
After 4th pass:	2, 3, 5, 4, 6, 7, 8, 9
After 5th pass:	2, 3, 4, 5, 6, 7, 8, 9
After 6th pass:	2, 3, 4, 5, 6, 7, 8, 9

Shuttle Sort

a) **4, 7, 9, 6, 8, 5, 2, 3**

After 1st pass:	4, 7, 9, 6, 8, 5, 2, 3
After 2nd pass:	4, 7, 9, 6, 8, 5, 2, 3
After 3rd pass:	4, 6, 7, 9, 8, 5, 2, 3
After 4th pass:	4, 6, 7, 8, 9, 5, 2, 3
After 5th pass:	4, 5, 6, 7, 8, 9, 2, 3
After 6th pass:	2, 4, 5, 6, 7, 8, 9, 3
After 7th pass:	2, 3, 4, 5, 6, 7, 8, 9

b) **5, 8, 7, 2, 3, 9, 6, 4**

After 1st pass: 5, 8, 7, 2, 3, 9, 6, 4

After 2nd pass: 5, 7, 8, 2, 3, 9, 6, 4

After 3rd pass: 2, 5, 7, 8, 3, 9, 6, 4

After 4th pass: 2, 3, 5, 7, 8, 9, 6, 4

After 5th pass: 2, 3, 5, 7, 8, 9, 6, 4

After 6th pass: 2, 3, 5, 6, 7, 8, 9, 4

After 7th pass: 2, 3, 4, 5, 6, 7, 8, 9

Quick Sort

a) **4, 7, 9, 6, 8, 5, 2, 3**

After 1st pass: 4, 7, 6, 5, 2, 3, 8, 9

After 2nd pass: 4, 2, 3, 5, 7, 6, 8, 9

After 3rd pass: 2, 4, 3, 5, 6, 7, 8, 9

After 4th pass: 2, 3, 4, 5, 6, 7, 8, 9

b) **5, 8, 7, 2, 3, 9, 6, 4**

After 1st pass: 2, 3, 5, 8, 7, 9, 6, 4

After 2nd pass: 2, 3, 5, 8, 7, 6, 4, 9

After 3rd pass: 2, 3, 5, 6, 4, 7, 8, 9

After 4th pass: 2, 3, 5, 4, 6, 7, 8, 9

After 5th pass: 2, 3, 4, 5, 6, 7, 8, 9