British Informatics Olympiad Final

6 – 8 April, 2001
Sponsored by Data Connection

Analogue Gadgets

Gadgets that calculate have been around far longer than the digital computers that are common place today. Analogue watches which work on cogs and springs have a history that includes orrerys (clockwork models of the solar system) and the *Antikythera* mechanism, a calendrical computer (made of several dozen cogs) that dates from the first century BC. Other devices rely on variable physical quantities, such as light, water pressure or electrical potential.

For example, the *straw sorter*, a device for sorting n numbers. For each number take a straw to a length equal to the number. You then bundle the straws together and bring them down on a flat surface. The maximum number is the highest straw. All you have to do to sort the numbers is to repeatedly remove the tallest straw. It is important to consider the processing required before and after the 'analogue' step; each number need a prepared straw (so n steps) and the straws are removed one by one (another n steps). There are potentially fewer steps here than it would take on a modern computer, although the individual steps will take longer.

1. Sorting & Searching

A more useful straw device is as follows: Consider a set of cards, where each card has an identical row of marked positions. At each position, on each card, there is either a notch or a hole, the exact combination being unique for each card. These cards are stacked up together. A straw is inserted at one position, passing through each card, and then lifted. All the cards with a hole at that position are lifted up; the notched cards slip off the straw.

Question 1.1
(a) Suppose there are 2^k cards and more than k different positions. What is the maximum (and minimum) number of cards that might come out at any stage?
(b) How about if there k positions?

Question 1.2
When the straw is lifted it is possible for more than one card to be lifted up. Suppose we are interested in finding a card with a specific combination of holes and notches. Outline an algorithm, using the straw technique, for finding a specified card.

While the order in which the index cards are stacked does not effect such straw search methods, it is sometimes useful for the index cards to be ordered. Suppose the cards are numbered from 1 to 2^k. If card 1 was the only one with a hole at position 1, card 2 was the only one with a hole at position 2, etc... it would be possible to sort (any random order of the cards) by selecting the cards in reverse order and placing each one (as it is found) at the front of the pack.

Question 1.3
Suppose the holes and notches have been made to represent the each card's number in binary. For example, card 5 (binary 101) would have "notch hole notch" (with holes in any the spare positions to the left). Outline an algorithm for sorting these cards.

Question 1.4
Sorting is not restricted to numbered systems. Using two long metal rods, design a system for sorting different sized oranges.

2. Paths

A set of cities are connected together by roads of varying length. It is possible to get between any two cities by some combination of roads, although not every pair of cities have a road directly between them. A well known computer science problem is to find the shortest path between two specified cities.

An analogue solution is to build a string model. Each city is represented by a small ring. If there is a road between two cities, the two rings are tied together by a piece of string, whose length is proportional to the length of the road. To find the shortest path between two cities in the string model you simply pull the two rings representing the cities until the model becomes taut. The taut pieces of string are the roads on the shortest path. The other strings will hang, loose, below the taut path.

Question 2.1
Why must the taut pieces of string represent the shortest path?

Question 2.2
What (and how much) processing is required before the analogue (i.e. pulling) stage? How about after the analogue stage?

A similar problem is to find the longest path between two cities. This is, surprisingly, a much harder problem. Consider the following analogue approach. Build the same model as before and pull the two rings representing the cities until the model becomes taut. Continue to pull the model until one of the taut strings breaks. Keep pulling and breaking taut strings. Just before the models falls into two pieces the longest path will be the one that is taut.

Question 2.3
This longest path algorithm does not always work. Why not? Illustrate with an example.

Consider the following variation on the problem. The cities are still connected together by roads, but the combination of roads is such that there is never more than one route (i.e. road or sequence of roads) between two different cities. In this case both the shortest and longest path between two cities is the same, since there is only one path between them.

Question 2.4
Using your string model of this new network, design an algorithm that determines which two cities are furthest apart.

Question 2.5
For your algorithm, what (and how much) processing is required before, during and after the analogue stage?