Interview – one-to-one preparation

Reminder – Have pen and paper ready and verbalise what you are doing to the interviewer(s), showing as well if asked either on screen or on paper

Problem 1

Write an algorithm to allow the user to enter an integer number for the number of paper bags, and a second integer (which must be greater than the first) for the number of sweets. The program then tells the user whether it is possible to put an odd number of sweets in each bag.

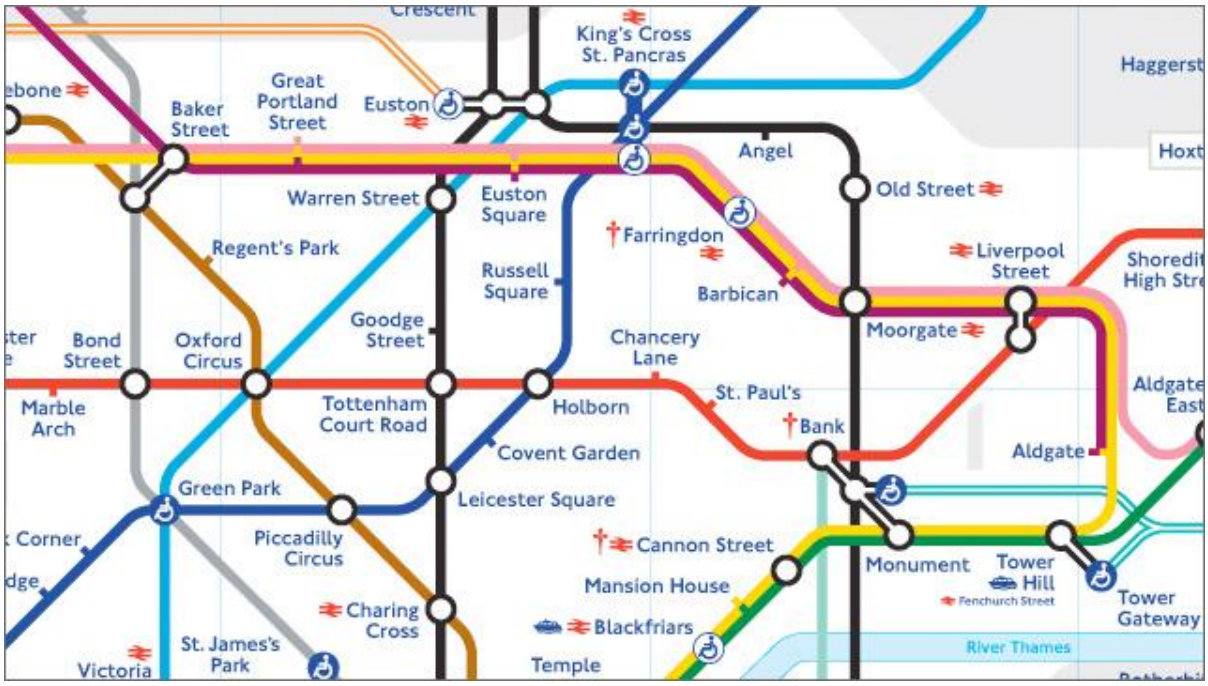
Problem 2

Monkey beans. An urn contains 23 white beans and 34 black beans. A monkey takes out two beans; if they are the same, he puts a black bean into the urn, and if they are different, he puts in a white bean from a large heap he has next to him. The monkey repeats this procedure until there is only one bean left.

What colour is it?

Discussion - Abstraction and reality

The London Underground map.



What information is shown, and what information is not shown? Does leaving out this information make the map clearer or less clear?

Work through exercise

A program has been written which contains a complex Boolean expression controlling a while loop.

Complete the truth table for the Boolean expression:

while ((a > b) or (b > c)) and ((not (a > b)) and (not (b > c)))

Let P represent a > b and Q represent b > c

P	Q	not P	not Q	P or Q	Not P and not Q	(P or Q) and (not P and not Q)
False	False					
False	True					
True	False					
True	False					

How many times will this loop be performed if a = 1, b = 2, c = 3?

How did you calculate your answer?

How many different permutations would have to be checked using a “brute force” method, on average, to crack a password, if it is known to be 4 uppercase alphabetic characters?

How many would have to be checked, in the worst-case scenario?

What is the Big-O time complexity of this algorithm?

A computer game has 5 doors which have to be opened in a particular sequence in order to progress to the next step. In how many different orders can the doors be opened?

What is the order of complexity of the problem if there are n doors?

Suppose there were 7 doors. How many doors would need to be opened, on average, to find the correct door using a "Brute Force" method?

Problem 3

Missing numbers. Imagine you are given a list of slightly less than 1,000,000 numbers, all different, and each between 0 and 999,999 inclusive. How could you find (in a reasonable time) a number between 0 and 999,999 that is not on the list?