A Decision Tree

E. L. Lady

Problem. Given 12 coins. Assume that at most one of them is defective, and that this one is either heavier or lighter than the others. Determine which coin is defective, using only three weighings by means of a balance scale. (C. f. Rosen, **Discrete Mathematics**, 4^{th} Edition, p. 541.)

This problem, which is essentially Problem 9 on p 546 of Rosen, is the classic problem of this sort and is much more challenging than the Example 3 on p. 541. It is an optimal problem in that the number of coins, 12, precisely matches what can be achieved with three weighings of a balance scale.

A common mistake in trying to solve this problem is to assume that the algorithm should use a number of conditional statements (**if** statements). For instance, one might start by weighing four of the coins against four others. If the pans balance, this would mean that all 8 of these coins are good, and one could then use these coins to test the remaining 4. On the other hand, if the pans don't balance, then one of the eight coins weighed must be bad, so use some other procedure whose goal is to isolate the bad coin. (Notice that in this second case, one knows that all four of the coins which were not weighed are good, and this might be useful.) Even so, it is hard to see how to manage with only two more weighings.

The standard solution, however, satisfies an additional constraint, namely that the choice of which coins be placed in which pan be predetermined in advance by the algorithm, independently of the outcomes of the three weighings. In other words, the algorithm might specify that on the second weighing, say, coin number 5 will always be placed in the left-hand pan, regardless of the result of the first weighing. This additional constraint, which seems so counter-intuitive, actually makes the problem much easier to analyse.

In any case, no matter what the algorithm is, it must be true that anyone who knows the algorithm and knows the results of the three weighings will be able to unmistakably identify the bad coin, if any, and say whether it is too heavy or too light.

Each weighing has three possible outcomes. Either the two pans balance (B), or the left-hand pan goes down (L), or the right-hand pan goes down (R). Thus the information available from the weighings can be coded as a three-letter sequence: RBL, for instance, or BBR. There are $3^3 = 27$ possible sequences of this sort. Each of these sequences (or at least the ones that actually occur) must serve to identify a single coin. But furthermore, each of the 12 coins can be either too heavy or two light, so this gives 24 possible situations that the weighings need to distinguish among. Furthermore, we have included the possibility that all coins are good, so the algorith has to be capable of actually producing 25 different answers. (The possibility that all coins are good clearly corresponds to the sequence BBB.) The 27 different outcome sequences are thus barely enough to identify all the possible situations. (It turns out that the sequences RRR and LLL will never occur in the algorithm we will use. In fact, in order for these sequences to occur, there would have to be one coin which is always placed in the left-hand pan or one which is always placed in the right-hand pan, and one can't make this work.)

The algorithm needs to specify which pan a given coin is put into on each of the three weighings. Since this choice is be independent of the results of the previous weighings, we see that the algorithm will assign a set of three instructions to each coin which can be coded as a sequence XYZ, where X, Y, Z must each be one of the three letters R (put coin in the right-hand pan), L (put coin in the left-hand pan), and B (do not put coin in either pan this time).

There are 3^3 (i. e. 27) possible instruction sequences, however the sequence BBB is not usable, because if a coin is left out of all three weighings and if all the other coins are good, we would have no way of knowing whether the BBB coin was bad or not or, if so, whether it was too heavy or too light.

Since there are 12 coins, there will be 12 different instruction sequences required (since clearly we can't do exactly the same thing to two different coins and still be able to distinguish which of the two is the bad one). It can also be seen that we should never use both a given sequence and its reverse as instruction sequences. Because if one coin has the instruction sequence RBL and another has the instruction sequence LBR, then we'd be unable to distinguish between the cases when the first coin was too heavy and the one when the second one too light. (In both cases, the right-hand pan would go down on the first weighing, the two pans would balance on the second, and the left-hand pan would go down on the third.)

We now assign each coin an instruction sequence XYZ, making sure only that we do not use both a given sequence and its opposite, and that we do not use the sequences BBB, RRR, or LLL for any coin. This leaves 24 allowable instruction sequences, of which we may only use half, which works out perfectly since there are exactly 12 coins.

However trying this scheme out shows that there is an additional requirement to be met.

For instance, if we assign the 12 coins the instruction sequences as follows:

RRL	RLL	BRR
RRB	RLB	BBR
RBL	BRB	LRB
RBB	BRL	LBB

we see that in fact none of these sequences is the opposite of any of the others. However this algorithm won't work, because there are 6 sequences of the form RXY and only two of the form LXY, so on the first weighing there are 6 coins in the right-hand pan and only 2 in the left-hand pan.

So we need to satisfy an additional requirement, namely that on each weighing there are an equal number of coins in each pan. In other words, the number of instruction sequences beginning with R should be the same as the number beginning with L, etc.

Whether the problem is solvable or not then comes down to the question of whether it is possible to find twelve instruction sequences satisfying this additional requirement.

Consider the letters R, L, B, in alphabetical order: B L R B, where we have added a second B at the end in order to make this ordering cyclic. Now let us consider the set of instruction sequences that either begin with a pair from this cycle, i. e. have the form BLX or LRX or RBX, or begin with a doubled letter together with a pair from this cycle, i. e. BBL or LLR or RRB. By symmetry, it seems that half the set of all possible instruction sequences (but excluding the sequences BBB, LLL, and LLL) should fit this pattern.

In fact, the instruction sequences in question are

BLR	LRB	RBL
BLL	LRL	RBR
BLB	LRR	RBB
BBL	LLR	RRB

A little thought (or else direct inspection) also reveals that it is never possible for both a sequence and its opposite to be in this list. (If a sequence has the form, for instance, BLX, then the opposite sequence would have the form BRY and hence would not be included in the list.) Finally, for reasons of symmetry we can see that one-third of the sequences in the list will have L in the first position (i. e. LXY), one-third will have R in the first position, and one-third will have B. Likewise for the second and third positions. So if we use these instruction sequences, then on each weighing there will be four coins in each pan and four coins left out.

It remains to find a way of determining the defective coin from the outcomes of the three weighings.

But now a little thought will show that the instruction sequence on the bad coin (if any) will be almost the same as the resulting outcome sequence. Namely, if a coin has, say, an instruction sequence RBL, and if this coin is too heavy, then on the first weighing (where the instruction sequence RBL shows that this coin is in the right-hand pan) the right-hand pan will go down, on the second weighing (where the B in the sequence RBL indicates that this coin will be left out of the weighing), the two pans will balance, and on the third weighing the left-hand pan will go down. So the outcome sequence will be RBL. Furthermore, no other coin will produce this outcome sequence if it is too heavy. On the other hand, if the coin with instruction sequence RBL is too light, then it will produce the opposite outcome sequence, namely LBR.

This essentially completes the algorithm. Start by labeling the 12 coins with the 12 instruction sequences in the set above. Then do three weighings, using the label on each coin to determine on each weighing whether it will be put in the left-hand pan (L), the right-hand pan (R), or neither (B).

If one of the coins is too heavy, then the outcome sequence for the three weighings will match the label of that coin. And if one of the coins is too light, then the outcome sequence for the weighings will be the opposite of the label on the coin. And if all of the coins are good, then the outcome sequence will be BBB.