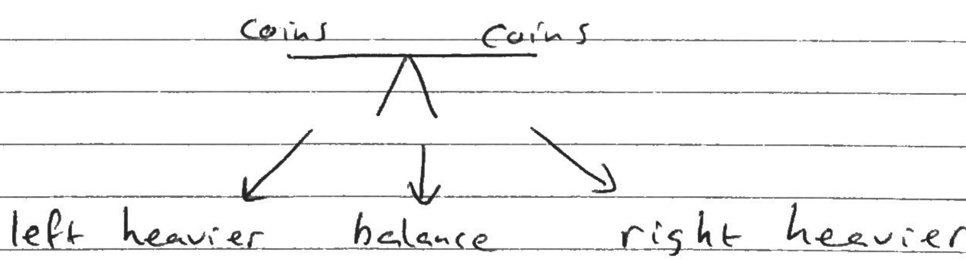
11.2 (continued)

We look at two more applications of trees.

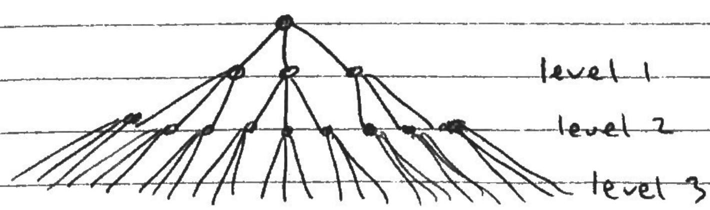
Decision trees

Complicated problems can be modeled and solved with the help of decision trees. They are rooted trees where each vertex corresponds to a decision (or result).

A classic problem gives an illustration. Suppose there are N gold coins. One is a fake though, weighing less than the others. You want to find the fake using a balance scale the fewest times. How many weighings do you need?



Every weighing has three possible outcomes so we get a 3-ary tree



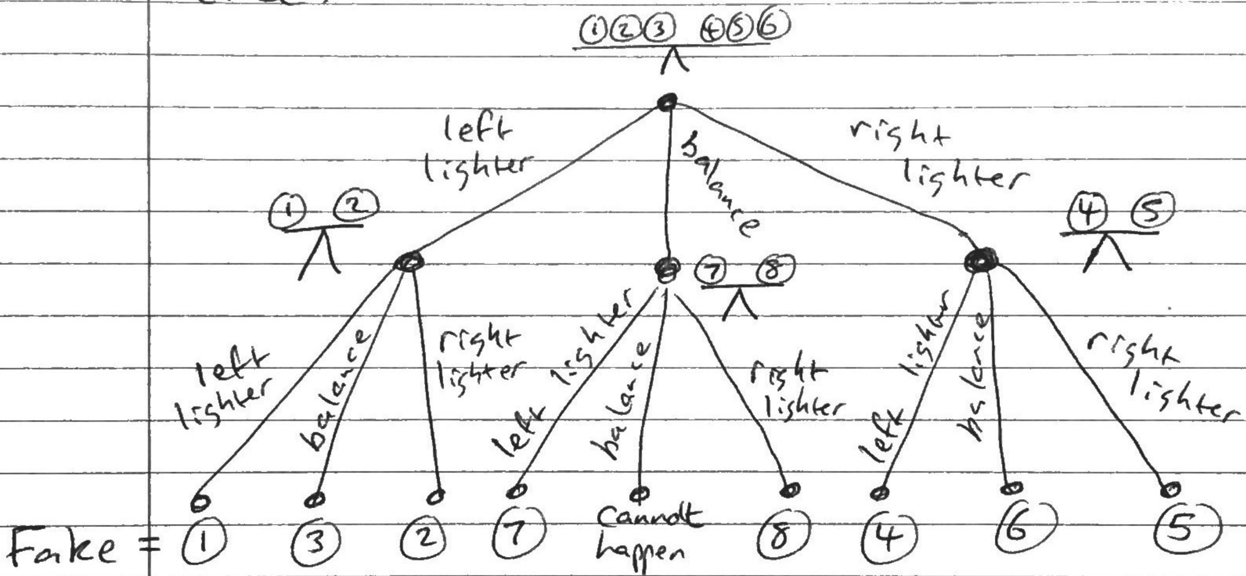
At level k there are at most 3^k vertices.

To find the fake in at most k weighings we need the 3-ary tree of height k to have at least N leaves. So we need $3^k \geq N$

or in other words $k \geq \log_3 N$.

So the smallest possible k is $\lceil \log_3 N \rceil$.

For example, with $N = 8$ coins, we need at least $\lceil \log_3 8 \rceil = \lceil 1.892 \rceil = 2$ weighings to find the fake.
It is possible to find the fake with just two weighings as seen in this decision tree:

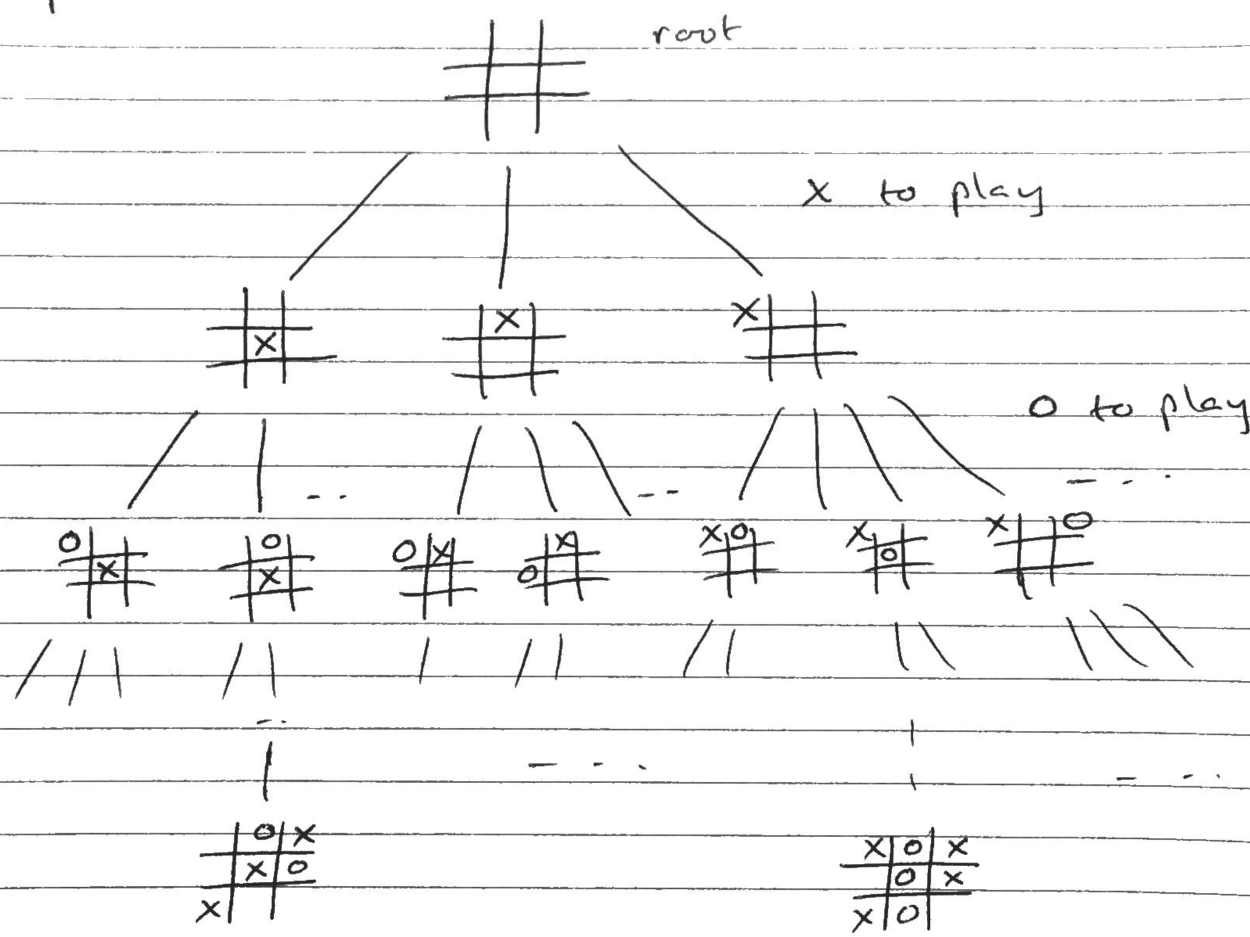


It can be seen that the same decision tree works for $N = 9$ coins.

(Generalizing these ideas a little, it can be shown that you can always find the lighter fake out of N coins with at most $\lceil \log_3 N \rceil$ weighings.)

Game trees (type of decision tree)

These are rooted trees that can be used to analyze 2-player games like tic-tac-toe, chess, nim. Each vertex is a game position (the root is the starting position) and the children are the next player's possible moves



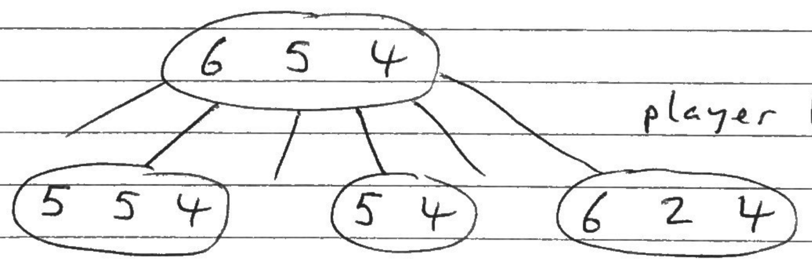
The game stops when someone gets 3 in a row. These positions are leaves. In this game leaves can also be draws.
The full game tree for tic-tac-toe has thousands of vertices.

The game chess has a much larger and more complicated game tree. Each position has usually over 20 children.

Nim

This is a simple game we can analyze. Two players play with piles of stones. On your turn you can remove any number of stones from one pile. The winner is the player removing the last stone. (Another way to say this is that you lose if there are no stones left and you cannot play — they have a slightly different version in the textbook.)

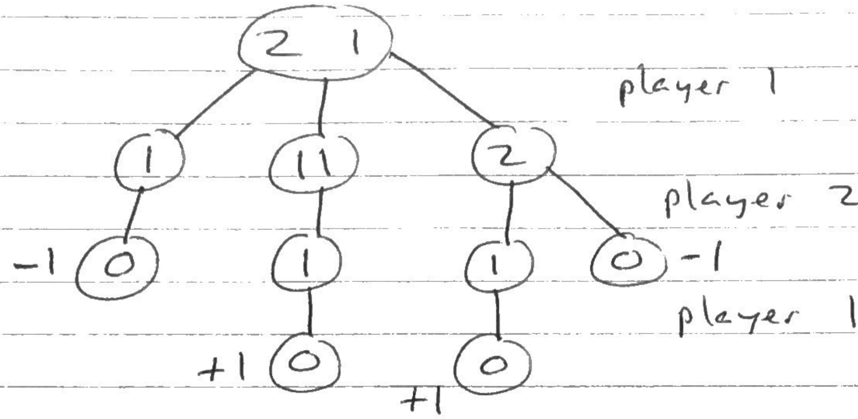
For example, suppose the game starts with 3 piles of 6, 5 and 4 stones.



The diagram shows three options for player 1 (the player going first). Which is the best move? Does player 1 have a winning strategy?

For a finite game like this, that must end in one player winning, it must be true that either the first or second player has a winning strategy.

Let's start with a simpler game where there are two piles of 2 stones and 1 stone:

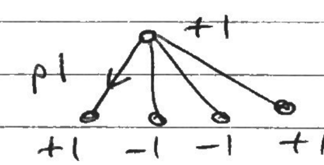


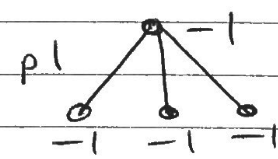
We have labeled the leaves with

$$\begin{cases} +1 & \text{if player 1 wins} \\ -1 & \text{if player 2 wins.} \end{cases}$$

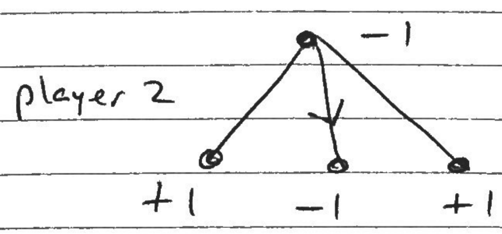
Now we can work back up the tree and label the rest of the vertices ± 1.

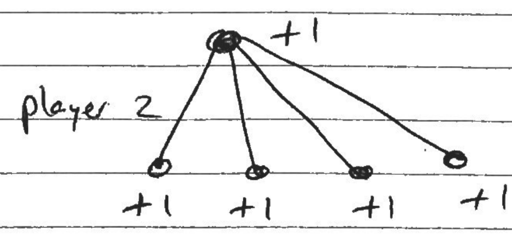
Player 1 should try to move to a +1 vertex on their turn. If they can the parent is labeled +1. But if they can't the parent is labeled −1.



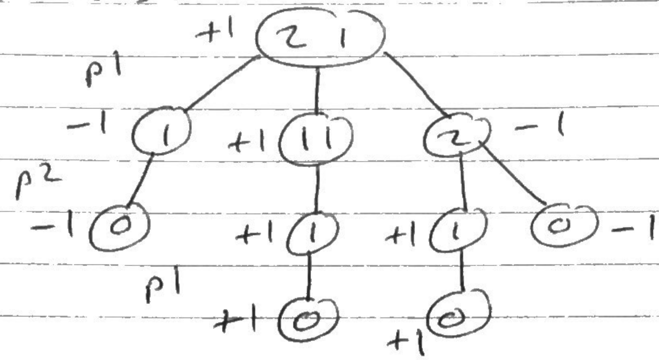


Similarly for player 2:





Using this procedure, we can fill in the +1, -1 values:



The starting position has value +1 so that means player 1 can always win (their winning strategy is to follow the +1s).

Here is the game tree for two piles of 2 stones each. It has a winning strategy for player 2.