Week Six

Note: The problems are taken from “Lab 8: Algorithm Problems” from the Addison-Wesley web resources from the text: Brookshear, J. Glenn 2005, Computer science: An overview, 8th edn, Addison Wesley.

Tutorial notes:

These problems are to be completed as groups. The aim is not only to solve the problem but to remember (and understand!) how you did so.

Problem 1: Coin Balance

This problem asks you to find one false coin among a set of true coins. The false coin can be found because it doesn't weigh the same as the rest - it is either lighter or heavier, but you do not know in advance. The only method you have to make a decision is a classic balance scale with two trays - you can put one or more coins on one tray, a similar number on the other tray, and determine which tray has the lighter pile. Say you only have three coins. Take coin 1 and coin 2 and weigh them. If the scale balances, then coin 3 must be the odd one out. If the scale doesn't balance, then compare coin 1 and coin 3. If they balance then coin 2 must be the odd one out, while if they don't then coin 1 must be the odd one out.

The key to the problem is to find the false coin using the fewest number of comparisons.

What is the fewest comparisons for 3 coins? For 4 coins? For 5 coins? For 12 coins? For N, an arbitrary number of coins? This problem is traditional stated with 12 coins.

Is there an algorithm that can always solve this problem?

Problem 2: The Thief’s Knapsack

In this problem a thief breaks into a vault that has a large collection of gems. The gems are of all types, from very rare diamonds to common stones. The thief has an experienced eye so she can instantly estimate the value and weight of each gem. In order to sneak away cleanly she carries a small knapsack that will only take N carats of gems. Her problem is to take the set of gems that in total are most valuable but that still fit in the knapsack. All weights are in integers (these are all large gems). She doesn't have to fill the knapsack exactly - it could be that the most valuable stones weigh less than N carats but she cannot add another gem without going over the weigh limit. (The general case of this problem is simply called knapsack.)

For example, there might be 7 gems, represented as pairs with weight, value: ruby (2 carats, $100); diamond (1 carat, $400); emerald (3 carats, $250); amethyst (2 carats, $50); opal (1 carat, $300); sapphire (2 carats, $500); malachite (1 carat, $60). The seven gems total 12 carats, but the thief can only carry 5.

Which stones should she take to get the maximum value in dollars? Is there an algorithm that can solve this problem?
**Task 1:**

Not all problems have solutions that can be neatly described with algorithms. Using the web, find some examples, be it “classic” or “real world”, problems that are not able to be solved with algorithms. What methods are used to solve these problems?

**Discussion point 1:**

Should people or corporations be able to “own” algorithms? What are the implications of ownership?