FIT1045 Algorithmic Problem Solving – Tutorial 3.

Objectives

The **objectives of this tutorial** are:

- To get familiar with bit lists.
- To understand *selection sort* and *insertion sort*.
- To introduce the fake coin problem

Task 1: Fake Coin Game

The Fake Coin Game This is a game with two players. Divide into pairs and decide who is the first player and who is the second player. You have 8 coins, numbered $1, 2, \ldots, 8$. One of the coins is a fake and is known to be lighter than the other coins. The first player decides which coin is a fake without telling the second player. The second player must now determine which coin is a fake by doing a series of weighings. For each weighing the second player says which coins has been put in each of the pans on the scale. The first player then informs the second player, after each weighing, whether the scale has tipped to the left, right, or has remained balanced. After the second player has guessed correctly, the players swap roles. The winner is the player who uses the smallest number of weighings.

Can you find an algorithm that uses the least number of weighings?

Task 2: Extended Fake Coin Game

We now add two complications to the Fake Coin problem:

- We now have 12 coins.
- We do not know if the fake coin is lighter or heavier.

How many weighings are necessary to find the fake coin and determine if it is lighter or heavier?

Task 3: Sorting

In small groups, sort six cards numerically using selection sort and insertion sort.

Describe the sequence of changes to the list $[P, Y, T, H, O, N]$ as it is sorted alphabetically using *insertion sort* and *selection sort*.

Task 4: Anagrams

Given a list of English words, describe an algorithm which can find all the anagrams for a given word. For example, if you were given the word *post*, you might return the list: [*pots*, *spot*, *tops*]. [1]

[1] A similar algorithm is probably behind: `http://www.wordsmith.org/anagram/`

Task 5: Bit lists

- Construct a list consisting of all the bit lists representing the subsets of $S = \{a, b, c\}$. Order the list in a way such that every list of bits differs from its immediate successor by one single bit. [2]
- Given a set S and two subsets of it, $A \subseteq S$ and $B \subseteq S$; design an algorithm to find the subset of S given by $A \cap B$. [3]

Puzzle of the week

The King of a small country invites 1000 senators to his annual party. As a tradition, each senator brings the King a bottle of wine. Soon after, the Queen discovers that one of the senators is trying to assassinate the King by giving him a bottle of poisoned wine. Unfortunately, they do not know which senator, nor which bottle of wine is poisoned, and the poison is completely indiscernible. However, the King has 10 prisoners he plans to execute. He decides to use them as taste testers to determine which bottle of wine contains the poison. The poison when taken has no effect on the prisoner until exactly 24 hours later when the infected prisoner suddenly dies. The King needs to determine which bottle of wine is poisoned by tomorrow so that the festivities can continue as planned. Hence he only has time for one round of testing. How can the King administer the wine to the prisoners to ensure that 24 hours from now he is guaranteed to have found the poisoned wine bottle?
Note: Assume no prisoner dies from natural causes in that 24 hours.

[2] Could it be useful to think about graphs in this problem as well?
[3] Assume that subsets are represented as lists of bits.