CIS 32 Tutorial 2

1. This is known as the *dogs & sheep* problem:

   > A farmer has three dogs and 3 sheep that he wants to get across a river. He has a boat, which in addition to him, will hold one or two animals, but no more. He can never leave more dogs than sheep either in the boat or on either side of the river.

   Writing $(s_1, d_1, k_1), (s_2, d_2, k_2)$ to indicate that there are s_1 sheep, d_1 dogs, and k_1 boats on the original side of the river and s_2 sheep, d_2 dogs, and k_2 boats on the far side of the river:

   (a) Write down the initial state and final state.

   (b) Write down the legal operations.

   (c) Draw out the state-space expanded by breadth-first search to at least 6 levels.

   (d) Give the optimum solution to the problem.

2. In chess, the average branching factor is 35.

   - What will the average size of an agenda be for a breadth first search in chess at depth 10?
   - Generalise your result from the first part to give an expression which gives the average size of an agenda in breadth first search in a problem with branching factor b at depth d.
   - Repeat the first two parts of this question for depth first search.

3. The *Towers of Hanoi* problem:

   > In a Tibetan monastry, there are 3 columns and 64 golden rings. The rings are of different sizes and rest over the

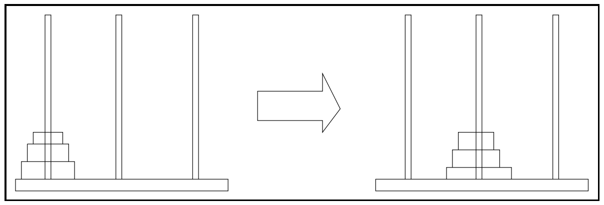

Figure 1: The 3 Ring Towers of Hanoi Problem

   (b) No ring may rest upon a smaller ring.

The 64 ring Hanoi problem is hard (and slow). But consider a small version of the problem using 3 rings (see Figure 1), draw out the full search space down to depth 3, and find one solution.

If you get stuck, you can find an implementation of a solver for this problem at:

```
http://eluzions.com/Puzzles/Java/Hanoi/
```

Note that this solver will give you some hints as to how to obtain a solution, but won't help you expand the whole search space.

4. Consider the Towers of Hanoi problem again. Write down the search tree for the 3-ring problem using:

   (a) depth limited search to depth 3;
   (b) iterative deepening to depth 4;

   Next, formulate a path cost function and an admissible heuristic for the problem, and solve it using:

   (a) uniform cost search;
   (b) greedy search;