

## **Problem 1.A**

### **Small Towers of Hanoi**

**Performance measure:** Move the disks from one rod to another without breaking the rules, make it as fast as possible. The disks should be put on rod 2/3 in a order of D1, D2, D3 (top to bottom).

**Environment:** Environment in this situation includes three rods and three disks.

**Actions:** Move a disk from one rod to another, and make sure no rule is violated.

**Sensors:** 1. Counter to keep track of how many time disks are moved  
2. Size checking to make sure the move is valid.

All rods can be represented by any dynamic arrays, while disks can be taken as object Disk, the object carries it's size and a method judging whether the incoming move is valid for this "disk".

### **Forward-Backward Search**

**Performance measure:** Travel through the map to eat all the pellets.

**Environment:** The rectangular grid, and pellets.

**Actions:** Move up, down, right and left, stop when hitting a wall, eat

when going through a pellet.

**Sensors:** 1. Sensor to check how many pellets are left, when it hits 0, the game ends. 2. Sensor counts how much time was used. 3. Step count sensor.

The environment of the Pac Man game can be a matrix similar to the Sliding Box Puzzle; walls are the slots that can't step on while pellets are special slots that can be found in a list. Once the game starts, the Pac Man list all the possible one-step ahead movements and perform one of the options. After that, it continues to do this recursively until it hits a spot where there is no way to go, then it goes back and find a new rout. Which is basically a DFS process.

## **Problem 1.B**

**Answer:** Showing that Franklin is one of Eloise's ancestors is more efficient than showing that Eloise is one of Franklin's descendants.

**Prove:** Assume Franklin have  $n$  descendants and we use a tree structure to store the information. To show that Eloise is one of the descendants of Franklin, we have to perform DFS or BFS search in order to achieve

the best time complexity  $O(n \log n)$ . When finding whether Franklin is one of the ancestors of Eloise, the time complexity is only  $O(\log n)$ , because each node has only one direct ancestor.