hw1

张芷苒 PB21081601

- 3.7 (a, b, d)
- 3.9

> **3.7** Give the initial state, goal test, successor function, and cost function for each of the following. Choose a formulation that is precise enough to be implemented.
> a. You have to color a planar map using only four colors, in such a way that no two adjacent regions have the same color.
> b. A 3-foot-tall monkey is in a room where some bananas are suspended from the 8-foot ceiling. He would like to get the bananas. The room contains two stackable, movable, climbable 3-foot-high crates.
> c. You have a program that outputs the message “illegal input record” when fed a certain file of input records. You know that processing of each record is independent of the other records. You want to discover what record is illegal.
> d. You have three jugs, measuring 12 gallons, 8 gallons, and 3 gallons, and a water faucet. You can fill the jugs up or empty them out from one to another or onto the ground. You need to measure out exactly one gallon.

a. **Map Coloring**

- **Initial State**: A planar map with all regions uncolored.
- **Goal Test**: All regions are colored with no two adjacent regions sharing the same color.
- **Successor Function**: Color an uncolored region with one of the four colors, ensuring no adjacent regions share the same color.
- **Cost Function**: Typically, the cost is constant, such as 1 for each coloring action since the goal is to achieve the correct coloring rather than minimizing the number of colors used (given the constraint of using only four colors).

b. **Monkey and Bananas**

- **Initial State**: Monkey is on the floor, crates are unstacked, and bananas are hanging from the ceiling.
- **Goal Test**: Monkey reaches the bananas.
- **Successor Function**: Move a crate, stack a crate on top of another, climb on or off a crate, or move the monkey on the floor.
- **Cost Function**: The cost could be defined as 1 for each action (move, stack, climb) to reflect the effort or time spent performing the action.

c. **Debugging Illegal Input Record**

- **Initial State**: A file of input records, with at least one causing an “illegal input record” error.
- **Goal Test**: Identify the record(s) causing the “illegal input record” error.

- **Successor Function**: Process the next record in the file or a subset of the records if using a divide-and-conquer approach.
- **Cost Function**: The cost could be measured in terms of processing time for each record or the number of records processed to find the illegal one.

d. **Water Jug Problem**

- **Initial State**: Three empty jugs and a water faucet.
- **Goal Test**: Exactly one gallon of water is measured out.
- **Successor Function**: Fill a jug from the faucet, pour water from one jug into another (either until the first jug is empty or the second jug is full), or empty a jug onto the ground.
- **Cost Function**: The cost could be constant for each action or variable, potentially based on the amount of water transferred to encourage efficiency in terms of water used or actions taken.

> **3.9** The **missionaries and cannibals** problem is usually stated as follows. Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place. This problem is famous in AI because it was the subject of the first paper that approached problem formulation from an analytical viewpoint (Amarel, 1968).
> a. Formulate the problem precisely, making only those distinctions necessary to ensure a valid solution. Draw a diagram of the complete state space.
> b. Implement and solve the problem optimally using an appropriate search algorithm. Is it a good idea to check for repeated states?
> c. Why do you think people have a hard time solving this puzzle, given that the state space is so simple?

a. **Problem Formulation**

- **State**: Represented by (Missionaries on the starting side, Cannibals on the starting side, Boat's side).
- **Initial State**: (3, 3, 0).
- **Goal State**: (0, 0, 1).
- **Successor Function**: Move 1 or 2 people to the other side without leaving more cannibals than missionaries on any side.
- **Cost Function**: Each move costs 1.

b. **Solving the Problem**

- **Search Algorithm**: Use Breadth-First Search for an optimal solution.
- **Repeated States**: Yes, checking for repeated states is important to avoid cycles and reduce search time.

c. **Difficulty for People**

- **Visualizing Constraints**: Keeping track of missionaries and cannibals and ensuring they're safe is not straightforward.
- **Memory Limitation**: Remembering past states and planning ahead is challenging due to limited working memory.
- **Non-Obvious Solution**: The solution requires strategic moves, including sometimes moving individuals back, which isn't immediately apparent.
- **Abstract Problem-Solving**: The puzzle demands abstract thinking and the ability to forecast future moves, skills that vary among individuals.