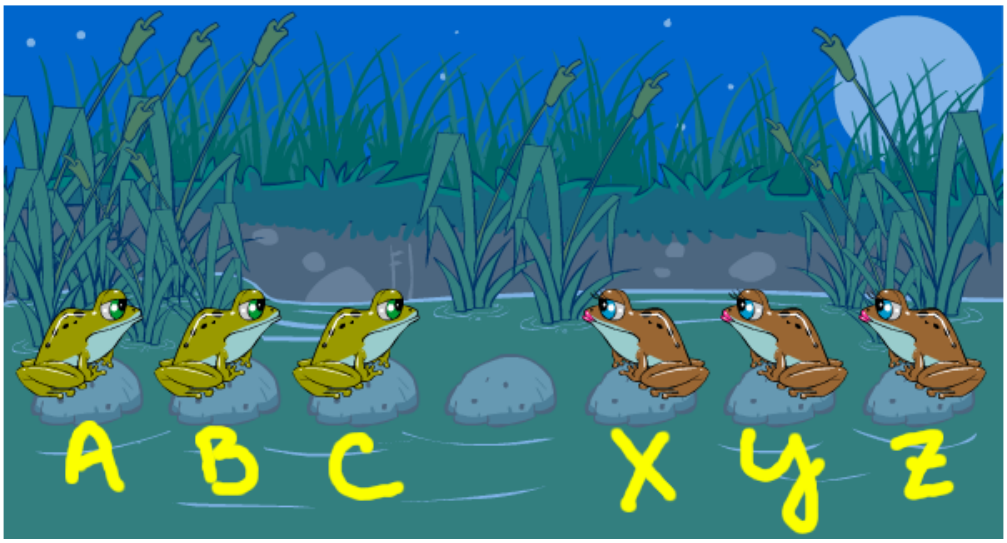
Assignment 10 – AI

SKIPPED

(1) Use search to solve a puzzle

Six frogs are trying to cross a pond by jumping between stones. Alma, Ben, and Carl are heading East; while Xavier, Yolanda, and Zanjoe are heading West. Each frog can jump just onto an adjacent stone, or jump **over** another frog if there is an empty stone behind it. The frogs are stubborn, however, and are **not** willing to change direction.



We can represent the start state above as `ABC-XYZ`. Then the goal state would be `XYZ-ABC`. From the start state, there are four possible moves. This chart emulates the first part of the search graph:

- `C` slides: `AB-CXYZ`
  - Then `B` slides: `A-BCXYZ`
    - Then `A` slides: `-ABCXYZ` and we're stuck
  - or `A` jumps: `-BACXYZ` and we're stuck
  - or `X` jumps: `ABXC-YZ`
    - Then `C` slides: `ABX-CYZ` ...
    - or `Y` slides: `ABXCY-Z`...
    - or `Z` jumps: `ABXCZY-` and we're stuck
- `B` jumps: `A-CBXYZ`
  - Then `A` slides: `-ACBXYZ` and we're stuck

- X slides: `ABCX-YZ` …
- Y jumps: `ABCYX-Z` …

On paper, expand the graph for the frog-jumping problem until you reach the goal state, but you can make two simplifications:

1. Just use four frogs, so the start state is `AB-YZ` and the goal is `YZ-AB`.
2. There are always two ways to reach the goal: one starts with a frog (A or B) moving right, the other with a frog (Y or Z) moving left. So ignore the solutions starting with Y or Z. This will cut the search space in half.

Here's a simple notation you can use:

```
ab-yz
  b: a-byz
  a: -bayz
  y: (ignored)
  z: (ignored)

a-byz
  a: -abyz
  y: ayb-z

-bayz
  stuck
```

I wrote a Frogs Python program that will solve this puzzle for any number of frogs.

(2) machine learning of decision tree

- Space shuttle data – after opening (use IE or Chrome, **not** Firefox), log in to Google and select **File » Make a copy.** Then you should be able to filter the records by field values using the drop-downs in the column header.
- Entropy calculator
- Decision tree assistant