

# Forge 3

**Due:** 11:59pm Sunday, February 23, 2020

**Please update** your Forge version to the newest version (when you run, it should say version 0.0.11). You will not be able to complete this assignment if you have not updated Forge.

Your solutions for this assignment should terminate within 30 or so seconds (don't panic if it takes longer! Please reach out to a TA though).

## Stencil Files

We have provided four stencil files for the four problems. They can be found in </course/cs1950y/pub/forge3>. You **must** follow the stencil.

## Problem 1: Stack

Model a last-in first-out [stack](#) in Forge with support for **Push** and **Pop** operations. In the stencil provided, each of the operations is an event. The stack should begin empty, and you should not be able to **Pop** from an empty stack (**Pop** should only ever operate on a non-empty stack). Note that duplicate elements are allowed in a stack.

What properties should a stack obey? You should write checks to verify your model's behavior in the following cases. Think about what should (or shouldn't) happen in each of them, and encode these each in their own check statement.

- A **Pop** followed by a **Push** of the same element
- The number of **Push** events is exactly equal to the number of **Pop** events
- A **Pop** from a stack with no elements (check that this does not occur)

## Problem 2: River Crossing

The following two problems are similar to a famous river crossing puzzle involving a goat, a wolf, and a cabbage.

### Part 1: Goats and Wolves

Three goats and three wolves are trying to cross a river in a boat that has room for at most two passengers. The boat cannot cross the river without any passengers. If there are ever more wolves than goats on one side of the river, the wolves will eat the goats. How can the goats and wolves cross safely to the other side? Solve the problem by modeling it in Forge and running it to find a solution. Remember, it's generally preferable to use as few constraints as possible to adequately constrain your model.

### Part 2: Pets and Owners

Three pet owners and their pets are trying to cross a river. Each owner has exactly one pet. Again, their boat has room for at most two passengers and cannot cross the river without passengers (both pets and owners are capable of driving the boat). If a pet is in the presence of another person, its own owner must also be present - otherwise, the pet's owner will be jealous and worry that the other person will try to steal their beloved pet! How can the pets and pet owners cross the river without making any of the pet owners jealous? You should solve the problem by modeling it in Forge and running it to find a solution.

### **Problem 3: Bridge Crossing**

Four people want to cross a bridge at night, but the bridge is narrow and at most two people can cross at once. They only have one torch, and due to the darkness, the torch must be always carried while crossing the bridge. Person A can walk across the bridge in 1 minute, B in 2 minutes, C in 5 minutes, D in 8 minutes, and when two people cross the bridge together, they must move at the slower person's pace. How can they all get across the bridge in 15 minutes or less? Model the problem in Forge using the stencil we provide, and run it to find a solution.

### **Documentation**

We've compiled some Forge documentation that might be helpful when doing these problems. You might find the [state transition](#) section helpful, as well as the [integer](#) section for the bridge crossing problem. You can find it [here](#).

We are still actively adding to it, so if you feel something is missing that should be included, let us know! We'll try to include it when we can.

### **Handing In**

Run `cs1950y_handin forge3` from a directory holding your 4 Forge files (`stack.rkt`, `goats_and_wolves.rkt`, `pets_and_owners.rkt`, `bridge.rkt`). You should receive an email confirming that your handin has worked.