CS4410 Homework 3

Due 9am Tuesday February 24

100 points as shown below

Short answer questions, 5 points each (no answer should need more than two or three sentences).

1. Name three ways in which the processor can transition into kernel mode.

2. What needs to be saved and restored on a context switch between two threads in the same process?

3. What is the difference between deadlock prevention and deadlock avoidance? What category does Bankers algorithm fall in and why?

4. Suppose you have a concurrent system with locks: `Lock.acquire()` blocks until the Lock is available and then acquires it. `Lock.release()` releases the Lock. There is also a `Lock.isFree()`, that does not block and returns true if the Lock is available; otherwise, returns false.

   Is the following code guaranteed never to block?

```
if(Lock.isFree())
    Lock.acquire();
```

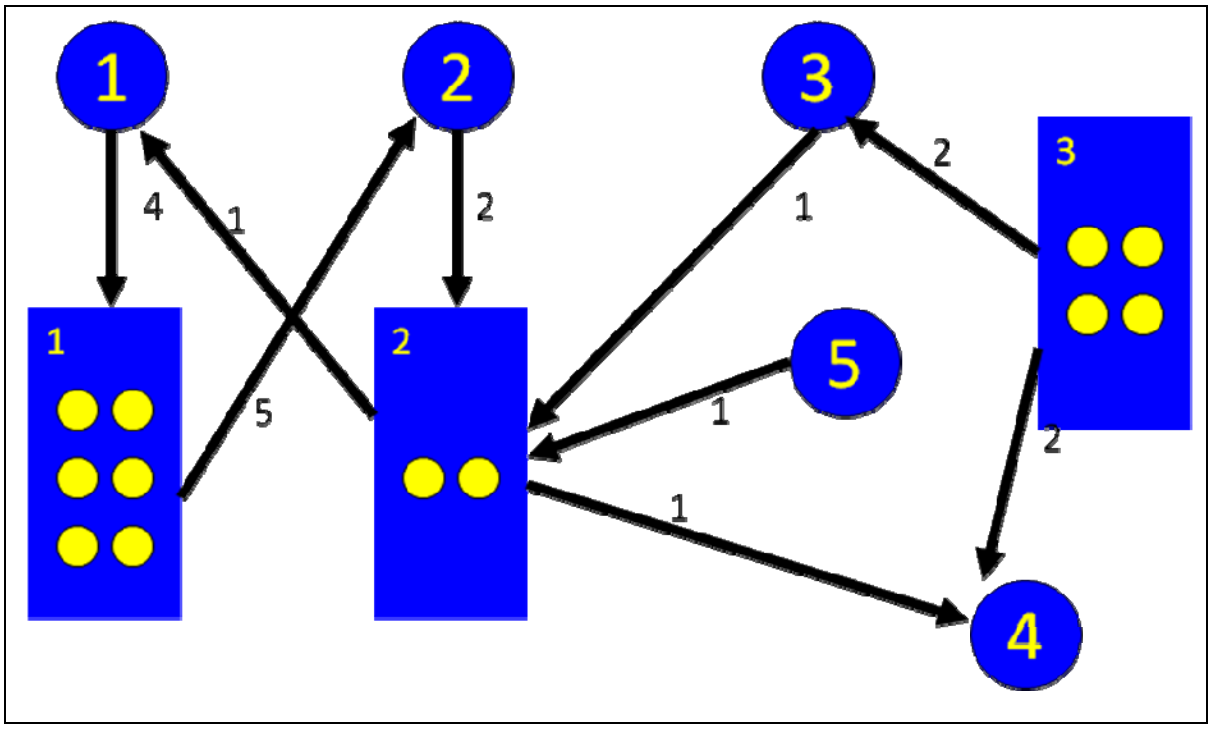
Ithaca's one-lane bridges

Write a class in Java to coordinate cars trying to cross one-lane bridges. The class should be instantiated once per bridge. A car wishing to cross the bridge will call bridge.cross(int direction), where direction is either 0 or 1, representing the two possible directions. This method blocks the car until it is safe for it to cross the bridge. Having finished crossing, the car will call bridge.done(). *Only one car can be on the bridge at a time.*

5. (25 points) Code a solution that solves the problem without worrying about fairness.

6. (25 points) Give a solution that will allow a maximum of 3 cars in a row to cross in any given direction and then switches to the opposite direction. It should only behave this way when there is actually "two way traffic" – your solution should not *assume* that there are cars trying to cross in both directions.

7. (30 points) For the resource-wait graph shown below, either show a graph-reduction sequence that fully reduces the graph, or prove that the system is deadlocked by showing us an irreducible subgraph.



Notation reminder: a process Pi is a blue circle with the number i inside, in yellow. Resource Rj is a blue box, with the value of j in the top left corner. The maximum total number of units of a resource is shown as a number of yellow circles. For example, below, there are a total of 4 units of R3. An arrow from a process to a resource, labeled, means the process is requesting that many units of the resource: P2 "wants" 2 units of R2. An arrow from the resource to a process means that the process owns that many units of the resource: P3 and P4 each own 2 units of R3. Notice that to figure out how many units of a resource are available, you'll need to sum up the labels on out-edges from that box and subtract this from the number of units shown inside the box. For example, all of R3 is currently allocated.