Lab 7

- the `Stack` in `getPath` is for reversing – don't change it to `Queue` or `PriorityQueue` for #3, #4!
  - tracing the "discovered from" info back from the goal yields the rooms on the solution path in reverse order – want path from start to goal

```
/**
 * Get the solution path, from the start to the goal. Requires that the solver
 * be done.
 *
 * @return solution path (the list is empty if the maze is unsolvable)
 */
@Override
public List<MazePos> getPath () {
        if ( !isDone() ) {
                throw new IllegalArgumentException("solver isn't done!");
        }

        // TODO [#2] create an empty Stack (holding MazePos) to hold the rooms found
        // on the path

        // find the rooms on the solution path from the "discovered from"
        // information
        for ( MazePos current = maze_.getGoal() ; current != null ;
        // TODO [#2] update current to be the room current was discovered from
        ) {
                // TODO [#2] add current to the rooms on the path
        }

        // TODO [#2] create an empty List (holding MazePos) to hold the path

        // TODO [#2] one-by-one, remove all of the rooms from the stack and add them
        // to the path list

        // TODO [#2] return the path
        return null;
}
```


Lab 7

- `Stack`, `Queue`, `PriorityQueue` operations
  - for various reasons, there are multiple methods for adding and removing
  - for `Stack`, prefer `push()` and `pop()`
    - these are the traditional names for stack operations, so using them makes the code clearer
  - for `Queue` and `PriorityQueue`, prefer `add()` and `remove()`
    - these are more traditional names for adding and removing, so using them makes the code clearer
    - `offer()` and `poll()` return special values instead of throwing exceptions if the queue is full or empty – it is easy to let those go unchecked, making bugs harder to find

Lab 7

- instance variables should be `private`
- declare variables of the most general applicable type
  - `Stack`, `Queue`, `PriorityQueue`
  - `Set`, `Map`