Describe and analyze an efficient algorithm that determines if a given maze has a solution.

**Solution:** We'll build a directed graph to model our progress through the number maze and then do a search to determine if the maze is solvable.

Let $G' = (V', E')$ be the graph we're going to build. Let each position in the grid be labeled as a pair $(i, j)$ meaning the position is in the $i$th row from the top and $j$ column from the left. We add a vertex $(i, j)$ to $V'$ for each pair where $1 \leq i \leq n$ and $1 \leq j \leq n$. For each vertex $(i, j) \in V'$, for each of the up to four positions $(i', j')$ we can reach from $(i, j)$ in a single move, we add an edge $(i, j) \rightarrow (i', j')$ to $E'$.

Each walk $(1, 1) \rightarrow (i, j)$ corresponds to a sequence of moves starting from $(1, 1)$ and going to $(i, j)$ and vice versa. Therefore, we want to know if $(n, n)$ is reachable from $(1, 1)$. We call `WHATEVERFIRSTSEARCH((1, 1))` and return `True` if and only if $(n, n)$ is marked by the procedure.

If we use a stack or queue as the “bag”, then the search takes $O(V + E)$ time. We have $|V| = n^2$ and $|E| \leq 4n^2$. **The algorithm takes $O(n^2)$ time.**

**Rubric:** 10 points total: 6 points for the algorithm. 2 points for the proof of correctness (mainly saying something about what the vertices, edges, and/or walks represent). 2 points for running time analysis.
Describe and analyze an algorithm to determine if there is a walk in $G$ from $s$ to $t$ (possibly repeating vertices and/or edges) whose length is divisible by 3.

**Solution:** We’ll build a separate directed graph $G' = (V', E')$ that models our progress walking through $G$ in a way that acknowledges how many edges we’ve used ($\mod 3$).

For each vertex $v \in V$, for each $z \in \{0, 1, 2\}$, we add a vertex $(v, z)$ to $V'$. For each edge $u \rightarrow v \in E$, for each $z \in \{0, 1, 2\}$, we add an edge $(u, z) \rightarrow (v, z + 1 \mod 3)$ to $E'$.

Observe, there is a walk in $G$ from $u$ to $v$ with $z \mod 3$ edges if and only if there is a walk from $(u, 0)$ to $(v, z)$ in $G'$. Therefore, we want to know if $(t, 0)$ is reachable from $(s, 0)$. We call `WHATEVERFIRSTSEARCH((s, 0))` and return `True` if and only if $(t, 0)$ is marked by the procedure.

If we use a stack or queue as the “bag”, then the search takes $O(V' + E')$ time. We have $|V'| = 3|V|$ and $|E'| = 3|E|$. **The algorithm takes $O(V + E)$ time.**

**Rubric:** 10 points total: 6 points for the algorithm. 2 points for the proof of correctness (mainly saying something about what the vertices, edges, and/or walks represent). 2 points for running time analysis.