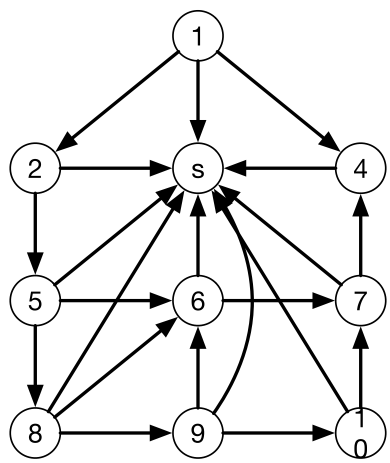
# Extra Credit Homework

Up to 100 points, that can be translated into either
    20 points for missing midterm and final score points
    100 points for homework and programming assignment points

**No collaboration, No use of the web!**
**Violators will have to face final judgement (or me if I find out).**

An *emil* in a directed graph is a node $s$ such that $\forall v \in V - \{s\} \ : \ (v, s) \in E$ and $\nexists v \in V \ : \ (s, v) \in E$. This means that all other nodes have an edge towards $s$ but no nodes has an edge towards $s$. An example for a digraph with universal sink is given below:




Assume that a matrix is given by an **adjacency matrix** with a coefficient $m_{i,j} = 0$ is there is no edge from Node $i$ to Node $j$, and $m_{i,j} = 1$ if there is an edge from Node $i$ to Node $j$. The adjacency matrix of the above graph is given as

$$(m_{i,j})_{i,j \in \{1,2,\ldots,10\}} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Give a $\Theta(|V|)$ time algorithm to determine whether a graph, given by an adjacency matrix, has an emil. Explain why your algorithm works.