1. **Snakes and Ladders** is a classic board game, originating in India no later than the 16th century. The board consists of an $n \times n$ grid of squares, numbered consecutively from 1 to $n^2$, starting in the bottom left corner and proceeding row by row from bottom to top, with rows alternating to the left and right. Certain pairs of squares in this grid, always in different rows, are connected by either “snakes” (leading down) or “ladders” (leading up). Each square can be an endpoint of at most one snake or ladder.

![A typical Snakes and Ladders board.](image)

You start with a token in cell 1, in the bottom left corner. In each move, you advance your token up to $k$ positions, for some fixed constant $k$. If the token ends the move at the top end of a snake, it slides down to the bottom of that snake. Similarly, if the token ends the move at the bottom end of a ladder, it climbs up to the top of that ladder.

Describe and analyze an algorithm to compute the smallest number of moves required for the token to reach the last square of the grid.

2. Suppose you are given a set of $n$ jobs, indexed from 1 to $n$, together with a list of precedence constraints. Each precedence constraint is a pair $(i, j)$, indicating that job $i$ must be finished before job $j$ begins. Describe and analyze an algorithm that either finds an schedule for executing all $n$ jobs on a single processor, such that all precedence constraints are satisfied, or correctly reports that no such schedule is possible.

3. For the previous problem, describe and analyze an algorithm to determine whether there is a unique schedule for a given number of jobs that satisfies a given set of precedence constraints.