Worth: 2% **Due:** By 9:59pm on Monday 29 September

Remember to write your *full name* and *student number* prominently on your submission.

---

*Please read and understand the policy on Collaboration given on the Course Information Sheet. Then, to protect yourself, list on the front of your submission **every** source of information you used to complete this homework (other than your own lecture and tutorial notes, and materials available directly on the course webpage). For example, indicate clearly the **name** of every student with whom you had discussions, the **title** of every additional textbook you consulted, the **source** of every additional web document you used, etc.*

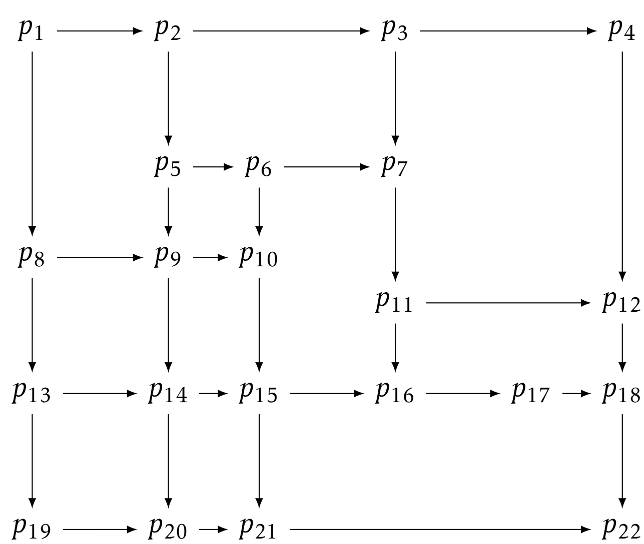
*For each question, please write up detailed answers carefully. Make sure that you use notation and terminology correctly, and that you explain and justify what you are doing. Marks **will** be deducted for incorrect or ambiguous use of notation and terminology, and for making incorrect, unjustified, ambiguous, or vague claims in your solutions.*

Consider the following problem, which we will call GRIDPATHS.

Input: A directed acyclic graph $G = (V, E)$, as in the example below. That is, the graph can be laid out on a grid so that all directed edges point either right or down, and no two edges cross. Further assume that $V = \{p_1, \ldots, p_n\}$ and that the vertices appear in order on the grid, when scanning left-to-right and top-to-bottom.

Output: The number of distinct directed paths from p_1 (the vertex at the top left corner) to p_n (the vertex at the bottom right corner).

For example, in the diagram below, there are three distinct paths from p_1 to p_{10} ($p_1 \to p_2 \to p_5 \to p_6 \to p_{10}$, $p_1 \to p_2 \to p_5 \to p_9 \to p_{10}$, $p_1 \to p_8 \to p_9 \to p_{10}$) but there is only one path from p_1 to p_{13} ($p_1 \to p_8 \to p_{13}$).



Use the dynamic programming paradigm covered in lecture to solve the GRIDPATHS problem. This problem is not particularly difficult to solve, so take the time to explain carefully what you are doing and to write up a detailed answer—in particular, follow closely the paradigm given in class.

Assume that you have at your disposal two arrays $up[i]$ and $left[i]$, defined as follows for $1 \leqslant i \leqslant n$:

$$up[i] = \begin{cases} j & \text{if there is a vertex } x_j \text{ such that} \\ & (x_j, x_i) \text{ is an edge going down,} \\ 0 & \text{otherwise.} \end{cases} \qquad left[i] = \begin{cases} j & \text{if there is a vertex } x_j \text{ such that} \\ & (x_j, x_i) \text{ is an edge going right,} \\ 0 & \text{otherwise.} \end{cases}$$

For example, in the graph above, $up[3] = 0$ and $left[3] = 2$, $up[11] = 7$ and $left[11] = 0$.