Worth: 2% **Due:** By 9:59pm on Monday 6 October

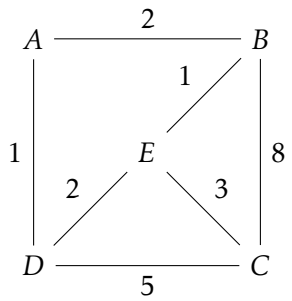
Remember to write your *full name* and *student number* prominently on your submission.

---

*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.*

It is possible for a graph to contain more than one minimum-weight path between two vertices u, v. For example, in the graph below, there is only one shortest path from A to B ($A \to B$) but there are three different shortest paths from A to C ($A \to B \to E \to C$, $A \to D \to E \to C$, $A \to D \to C$).



Use the dynamic programming paradigm covered in lecture (and in the textbook) to solve the following problem.

- **Input:** A graph $G = (V, E)$ with edge weights $w(e) \in \mathbb{N}$, for all edges $e \in E$.
- **Output:** For all $u, v \in V$, the **number** of distinct minimum-weight paths from u to v in G.

Briefly argue that your algorithm is correct (based on the recursive structure of the problem, which you should describe) and analyse its running time.

HINT: Base your algorithm on the the idea of restricting the number of edges on the shortest paths (as described in the tutorial for Week 4). This makes the solution easier to justify.