Worth: 12% **Due:** By 9:59pm on Wednesday 26 November

Remember to write the *full name* and *student number* of *every group member* 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 from another group 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.*

---

1. A *decimal counter* is a string of k digits $d_{k-1}d_{k-2}\cdots d_1 d_0$, where each $d_i \in \{0,1,2,3,4,5,6,7,8,9\}$.

   (a) Suppose that the only operation is INCREMENT and that the counter starts with all digits equal to 0. Use the *accounting method* to analyse the amortized complexity of any sequence of m operations. For full marks, make your charge as small as possible, and state and prove an appropriate *credit invariant*.

   (b) Now suppose that the counter supports two operations: INCREMENT (as before) and RESET (set every digit back to 0).

   Give a detailed analysis of the amortized complexity of any sequence of m operations. As before, make your charges as small as possible, and state and prove an appropriate credit invariant.

2. When booking rooms to hold the final exam, we want to make sure to book enough rooms so that we can split up classmates who might unfairly communicate with one another during the exam. Judging by who listed each other on the first page of their assignments and by who showed up to office hours together, we can form an undirected *collaborators graph* $G = (V, E)$, where V is the set of students and E is the set of unordered pairs of collaborators. For convenience (and to preserve privacy), we use the numbers $1, \ldots, n$ instead of actual student numbers.

   (a) Show that, for the collaborators graph represented by the following adjacency lists, there is a way of splitting up the students into two rooms such that no two collaborators end up in the same room.

   1: [3, 6]
   2: [4]
   3: [1, 4, 5]
   4: [2, 3, 6, 7]
   5: [3, 6, 7]
   6: [1, 4, 5]
   7: [4, 5, 8]
   8: [7]

Given the collaborators graph G, one algorithm for solving this problem considers each student one at a time and assigns the student to the first room that none of his/her collaborators have been assigned to. More specifically, suppose we have rooms numbered $R_1, R_2, R_3, \ldots$, which initially contain no students. Then the algorithm is:

AssignRooms(G):
  for $i \leftarrow 1, \ldots, n$:
    Find the minimum j such that R_j contains no neighbour of student i
    Add i to R_j

(b) Describe how you would implement AssignRooms using the adjacency lists representation of the graph G. Write your answer in pseudocode and fill in the details of how each step is carried out. Your implementation should run in worst-case time $\mathcal{O}(n)$.

(c) Describe how you would implement AssignRooms using the adjacency matrix representation of the graph G. Write your answer in pseudocode and fill in the details of how each step is carried out. Your implementation should run in worst-case time $\mathcal{O}(n)$.

(d) If you were trying to minimize running time, which representation would you choose and why?

(e) Trace the algorithm AssignRooms on the collaborators graph given in part (a). How many rooms does it use? Show your work.

(f) For any collaborator graph, let k be the minimum number of rooms we need so that we can split up the collaborators into separate rooms. Prove that there exists a numbering for the students such that running AssignRooms on the resulting collaborators graph uses k rooms.

3. Let $G = (V, E)$ be a connected, undirected graph.
   - An *articulation point* of G is a vertex whose removal disconnects G.
   - A *bridge* of G is an edge whose removal disconnects G.

   Let $G_\pi = (V, E_\pi)$ be a depth-first tree for G; in other words, $E_\pi = \{\{\pi[v], v\} : v \in V\}$.

   (a) Prove that v is an articulation point of G if and only if:
   - v is the root of G_π and v has more than one child in G_π; or
   - v is not the root of G_π and G_π contains **no** back edge (u, w) such that u is a descendant of v and w is an ancestor of v.

   (b) Prove that e is a bridge of G if and only if e does not belong to any simple cycle in G.

   (c) Modify DFS to return every articulation point and every bridge in an undirected graph G. Make your algorithm as efficient as possible and analyse its worst case running time.

   Hint: For each vertex v, compute a new field $m[v]$ equal to the *minimum* value of $d[u]$ over all vertices u such that u is an ancestor of v in G_π and there is some path from v to u in G. Use this information, together with the results you proved in part (a), to identify the articulation points.

4. Let $G = (V, E)$ be a connected, undirected graph, on which we run BFS starting from some vertex $s \in V$.

   (a) Show that $T = \{\{\pi[v], v\} : v \in V\}$ is a spanning tree of G.

   (b) Let each edge $e = \{u, v\} \in E$ be assigned weight $w(e) = d[u] + d[v]$—where $d[u]$ and $d[v]$ are the values computed during BFS. Prove or disprove that T is a minimum spanning tree.