1. A graph is called 3-regular (or ‘cubic’) if every vertex has degree 3. Prove that a 3-regular graph has an even number of vertices.

[Hint: use the Handshaking Lemma, given on p. 3 of the notes to Week 1, Lecture 3].

**SOLUTION:** Let \( G = (V(G), E(G)) \) be a 3-regular graph. The Handshaking Lemma tells us that

\[
\sum_{v \in V(G)} d(v) = 2|E(G)|.
\]

Now since \( d(v) = 3 \) for every vertex \( v \) of \( G \), this equality becomes \( 3|V(G)| = 2|E(G)| \) whence \( |V(G)| \) must be divisible by 2.

\[\square\]

2. In the Week 1, Lecture 3 notes, the following pseudocode was given to calculate the value of \( d(v) \) for a vertex \( v \):

\[
d(v) := 0
\text{for } e \in E \text{ do}
\quad \text{if } v \in e \text{ then } d(v) := d(v) + 1
\]

Amend this pseudocode to make it correct when the input graph can include self-loops.

**SOLUTION:** Self-loops at vertex \( v \) contribute 2 to \( d(v) \). So we must test if \( e \) is a self-loop, i.e. if \( |e| = 1 \). The most obvious amendment is:

\[
d(v) := 0
\text{for } e \in E \text{ do}
\quad \text{if } v \in e \text{ then}
\quad\quad \text{if } |e| = 1 \text{ then } d(v) := d(v) + 2 \text{ else } d(v) := d(v) + 1
\]

A slightly more devious alternative you might have thought of is:

\[
d(v) := 0
\text{for } e \in E \text{ do}
\quad \text{if } v \in e \text{ then } d(v) := d(v) + \frac{2}{|e|}
\]

3. Prove the Handshaking Lemma by induction on the number of vertices.

**SOLUTION:** We must show that the sum of the degrees is twice the number of edges for any graph \( G = (V, E) \).

**Base case:** we may start with \(|V| = 0\) (this seems slightly sneaky but it is always a good idea to start with the lowest possible value — you just have to take care not to fall into a ‘monochromatic cows’ trap! \(^1\)). Indeed, when \(|V| = 0\) we have \( 2|E| = 2 \times 0 = 0 = \sum d(v) \).

**Inductive Step:** Assume the result is true for \(|V| = K \geq 0\). Let \( G \) have \( K + 1 \) vertices. We choose a vertex \( u \) and delete it, together with all incidence edges. This gives us a graph \( G' = (V', E') \) on \( K \) vertices for which, by inductive hypothesis,

\[
\sum_{v \in V'} d(v) = 2|E'|.
\]  

\(^1\)E.g. see plus.maths.org/content/monochromatic-cows
Now replace the vertex \( u \) and replace one of the incident edges, say \( e \). Clearly this increases the right-hand side of equation (1) by 2. If \( e \) is a self-loop at \( u \) then the left-hand side of equation (1) is increased by 2. If \( e \) is not a self-loop then the left-hand side of equation (1) is increased by one for each of its endpoints. So again the left-hand side is increased by 2. So equation (1) remains true when we replace a deleted edge. Replacing all the deleted edges gives us the required result for graph \( G \).

The Handshaking Lemma now follows by induction. \( \square \)

4. The following pseudocode specifies an algorithm which checks if a graph \( G \) has an isolated vertex (that is, having zero degree).

Algorithm SeekIsolated

Input: graph \( G = (V, E) \)

Output: True if there is a vertex \( v \in V \) for which \( d(v) = 0 \); False otherwise

1. for \( v \in V \) do
   foundzero:=False # initialise to a False answer, then look for evidence to the contrary
   foundincidence:=False # if this stays False in the following loop then
   # we’ve found a zero degree
   2. for \( e \in E \) do
      if \( v \in e \) then
         foundincidence:=True
   3. if not foundincidence then foundzero:=True
      return(foundzero)
   end
end

Given an analysis of the complexity of this algorithm.

SOLUTION: The for loop at (1) takes \(|V|\) steps to examine each vertex.

For each vertex the for loop at (2) takes \(|E|\) steps to examine each vertex. After this loop finishes we take 1 step to test if not foundincidence.

So the total time is \(|V| \times (|E| + 1)\) steps. Since \(|V| \times (|E| + 1) = |V||E| + |V| \) and the 2nd term is less than the first we can write this as \( O(|V||E|) \) steps.

5. [FOR MARKING]

(a) In the Week 1, Lecture 3 notes the following Lemma was proved:

   **Lemma** If \( G \) is a simple graph on two or more vertices then \( G \) has two vertices with equal degree.

   Is this Lemma true for non-simple graphs? What about if self-loops are allowed but not multiple edges? What if multiple edges are allowed but not self-loops? Give counter examples as appropriate.

(b) We could speed up the algorithm Stupid, given on p. 4 of the Week 1, Lecture 3 notes, by only checking all \( \binom{|V|}{2} \) pairs of vertices. The for loop at 1. in the algorithm specification would still check every vertex but the for loop at 2. would only check vertices further down the list. What is the complexity of this new version of Stupid? Is it different from the original version when we use the big-Oh notation?
SOLUTION:

(b) Counterexamples for (a) non-simple, (b) loops only, and (c) multiple edges only are shown below: [2 marks each]

\[\begin{array}{ccc}
\text{(a)} & \text{(b)} & \text{(c)} \\
\begin{array}{ccc}
4 & 2 & 2 \\
3 & 1 & 3 \\
\end{array}
\end{array}\]

(b) The question does not require pseudocode for the new version of Stupid, but here it is for the sake of completeness:

Algorithm Stupid
Input: simple graph $G = (V, E)$ with $V = \{v_1, v_2, \ldots, v_n\}$
Output: True if there are vertices $u, v \in V$ for which $d(u) = d(v)$; False otherwise

Found:=False # initialise the result

1. for $u \in V$ do
   Let $u = v_i$, say.
2. for $v \in \{v_{i+1}, \ldots, v_n\}$ do
3. if $d(u) = d(v)$ then Found:=True

return(Found)

Complexity: the for loop at (1) examines each vertex but each iteration of the loop will require fewer steps: [4 marks]

- for the first vertex the for loop at (2) takes $|V| - 1$ steps to examine $v_2, \ldots, v_n$;
- for the second vertex the for loop at (2) takes $|V| - 2$ steps to examine $v_3, \ldots, v_n$;
- for the penultimate vertex the for loop at (2) takes 1 steps to examine $v_n$;
- for the last vertex the for loop at (2) will not execute (no vertices remain to check).

So this is a total of $|V| - 1 + |V| - 2 + \ldots + 2 + 1 = \frac{1}{2}|V|(|V| - 1)$ iterations of the for loop at (2). [4 marks]

OR: just observe that $\binom{|V|}{2}$ pairs of vertices to check is $\frac{1}{2}|V|(|V| - 1)$ checks. (also earns 8 marks).

Now each iteration of the for loop at (2) uses $2|E|$ steps to compare $d(u)$ and $d(v)$. [2 marks]

So the total time complexity is $\frac{1}{2}|V|(|V| - 1) \times 2|E| = |V|^2|E| - |V||E|$ steps. [4 marks]

The complexity of the original version of Stupid was $|V|^2 \times 2|E|$ steps. In big-Oh notation we ignore multiples so this is $O(|V|^2|E|)$ (as explained in the lecture notes).

For the new version we ignore the subtracted 2nd term since it is smaller than the first term. This gives $O(|V|^2|E|)$. [4 marks]

So the new version has the same complexity, using big-Oh. [1 mark]