Special instructions: You may work either individually or in a group of exactly two persons. Write your solutions out on paper, photocopy them and hand in both the original and the copy to SSO by 14:00 on Friday, 18th October, 2019. If you are working in a group, one script should be submitted for the group, with both names at the top; both members of the group will then receive the same mark. Clearly write your name(s), student ID number(s) and the words "36111-cwk2-S-exerciseA" on the front (cover) sheet and staple all sheets together. Submitted solutions must be entirely the effort of the persons whose names appear at the top of the submission.

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Advanced Algorithms I: 36111-cwk2-S-exerciseA

Time: This should take you a few hours

Please answer all questions.

The marks for this exercise WILL count towards your final mark for Comp36111.

The use of electronic calculators is <u>not</u> recommended.

Section A

All terms relating to graphs and directed graphs are to be understood as in the course slides. In particular, (directed) graphs are by assumption finite, have no self-loops and no multiple edges.

1. Show that every non-empty acyclic directed graph has at least one vertex with in-degree 0.

   (5 marks)

2. Show that a directed graph is acyclic if and only if it has a topological ordering.

   (5 marks)

3. Let $G = (V, E)$ be a directed graph. Define a *strict cycle* in G to be a sequence of distinct vertices $v_0, \ldots, v_{k-1}$ ($k \geq 2$) such that, for all i ($0 \leq i < k-1$), $(v_i, v_{i+1}) \in E$, and $(v_{k-1}, v_0) \in E$. Using the definitions in the slides, show carefully that G is acyclic if and only if it does not contain a strict cycle. (5 marks)

4. Define a graph to be *bi-partite* (some say "2-colourable") if there is a partition of V into (possibly empty) disjoint sets, $V = U_0 \cup U_1$ such that for every edge e, exactly one vertex of e is in U_0 (and hence exactly one vertex of e is in U_1). Consider the following modification of `DFS`:

```
begin bCheck(G = (V,E))
    take all v ∈ V to be uncoloured
    until all vertices in V coloured
        pick some uncoloured u ∈ V
        set colour(u) := 0
        if DFSb(G,u) = false
          return false
    return true
end
```

```
begin DFSb(G, u)
    for all v such that (u,v) ∈ E
        if v is not coloured
           set colour(v) := 1 - colour(u)
           if DFSb(G, v) = false
              return false
        if colour(u) = colour(v)
           return false
    return true
end
```

   Show that the following are equivalent:

   (a) G is bipartite;

   (b) G contains no cycle of odd length;

   (c) `bCheck`(G) returns `true`.

   (5 marks)

END OF EXAMINATION