Some Exercises and Answers on Advanced Algorithms

- In this question, by graphs we mean finite, undirected graphs.

  1. Explain clearly what is meant by a depth-first search (DFS) of a graph. Using suitable notation, write a program to perform DFS of graphs.
  2. Explain how a DFS of a graph separates the edges of the graph into several classes.
  3. What is meant by an articulation point of a graph?
  4. Explain clearly how DFS can be used to calculate the articulation points of a graph (you may either write a program - which you should explain - or give a clear explanation of how a DFS can be used to determine articulation points).

Answer. This is mainly a bookwork exercise but is included so as to indicated the scope and level of a typical exam question.

The algorithm expected for DFS of a graph is the recursive one given in the notes:

```
forall nodes u set dfsnum(u) = 0;
set i = 0;

visit(u) =
   { set i = i+1;
     set dfsnum(u) = i;
     forall nodes v adjacent to u
       { if dfsnum(v) = 0 then visit(v) };

forall nodes u
   {if dfsnum(u) = 0 then visit(u)};
```

For full marks you should give such pseudocode (or real code) and explain it.

For undirected graphs DFS has only TREE edges and BACK edges, which you should explain.

DFS algorithm for articulation points: You may either (1) give the proposition and proof in the notes explaining how calculating of low-point numbers for a DFS enables us to identify articulation points, or (2) modify the above recursive DFS to calculate lowpoint numbers and identify articulation points and explain the code.

- 1. Explain what is meant by an Eulerian circuit in a finite undirected graph.

     What simple property of the graph corresponds exactly to the existence of Eulerian circuits?

     Explain clearly how to compute an Eulerian circuit in such graphs (you need not write a program, but whatever algorithm you choose must be clearly explained).

  2. For finite undirected graphs, explain what is meant by a colouring of a graph.

     Show that if a graph can be coloured with two colours then there are no cycles of odd length.

     Show conversely that if there are no cycles of odd length then the graph can be coloured with two colours.

  Answer. The simple property equivalent to the existence of Eulerian circuits is that the graph is connected (ie consists of one connected component) and every node has even degree (ie number of edges at each node is even).

  To compute an Eulerian circuit in such a graph, proceed as follows:

  - Choose any start code. Trace an arbitrary path, never reusing an edge. This will terminate at the start node, because each node has even degree so once reached a node can always be left by an unused edge.
  - If all edges are used, then stop with an Eulerian circuit. Otherwise, the remaining edges will form a subgraph which consists of a number of separate components but each having all nodes of even degree. Moreover, because the original graph was connected, each of these subgraph components, will meet the partial Eulerian circuit at at least one node.
  - Recursively compute Eulerian circuits of these components (this is correct by induction, they are all smaller than the original graph).

Finally, expand the original partial circuit by each of these circuits to form a circuit of the whole graph.

Two-colouring. For the first part: Suppose that graph G does have an odd cycle. Then we must show that it cannot be two-coloured. Let the cycle be $n_1, ..., n_k$ where k is odd. Then for a two-colouring we alternate colours, and as k is odd, the colour of n_1 is the same as that of n_k, but as it is a cycle there is an edge between these nodes, so at least a third colour is required to colour G.

Conversely: We need to manufacture a two-colouring given that all cycles are even. To do this, we pick a node arbitrarily and use one colour (red say). Then all uncoloured nodes linked to this are coloured with the other colour (green say). Nodes linked to this green that are not already coloured are coloured red etc. This can only go wrong if there is an edge from between nodes of the same colour. In this case we show there must be an odd cycle, let m and n be the nodes which are linked and both coloured red, say. Now, there is a path from the start node to each of m and n and a simple diagram shows that these together with the edge between m and n must form an odd cycle. Likewise if the two nodes m, n are both linked and green.

- This question is about showing a problem on finite, undirected graphs is NP-complete.

  An independent set of nodes in a graph is a set of nodes no two of which are linked by an edge. The problem is to determine, for any graph, whether or not it has an independent set of nodes of size k.

  Show firstly that the problem is in NP i.e. that we can test in polynomial time whether a possible solution is a solution.

  Recall that the problem of determining whether a graph has a clique of k nodes is NP-complete.

  Now show that the problem of determining the existence of independent sets of size k is NP-complete by reducing the problem of finding cliques to the problem of finding independent sets of nodes. Hint: consider the complement of a graph G, ie the graph which has the same set of nodes as G but edges only between pairs of nodes which are not linked by an edge in G.

Answer. To show the problem is in NP, we need to show that a proposed solution can be tested in polynomial time. Now, a proposed solution is a set of k nodes in a graph G with N nodes. To test whether this is an independent set is simply to ensure that no pair of nodes in the set is linked by an edge. There are k^2 pairs of nodes to be tested and each test can be performed in constant time (for adjacency matrix) or linear time (for adjacency list look-up). In both cases, the overall test is polynomial.

Now, we need to translate the given NP-complete problem (finding a clique with k nodes) into the problem of finding independent sets of k nodes. As the hint suggests, consider a graph G, and its complement G*, which has an edge exactly where G doesn't. Construction of G* from G is a polynomial-time operation, considering all N^2 pairs of nodes, and for each checking whether there is an edge in G.

The key observation then is S is an independent set in G* if and only if S is a clique in G, by definition. So a clique of k nodes in G corresponds exactly to an independent set of k nodes in G*, as required. Thus the problem is NP-complete.

- This is another question about showing that a graph problem is NP-complete.

  We state as a fact that the problem of finding whether or not there is a Hamiltonian circuit in a directed graph is NP-complete.

  You are then asked to show that the problem of determining whether or not an undirected graph has a Hamiltonian circuit is NP-complete, as follows:

  Show firstly that the problem for undirected graphs is in NP i.e. that we can test in polynomial time whether a possible solution is a solution.

  Now show how to translate the problem for directed graphs into that for undirected graphs. Hint: consider the following construction of an undirected graph G' from a directed graph G. For each node n in G, there are three nodes n-IN, n-MID, and n-OUT in G'. For an edge from u to v in G, there is an indirected edge between u-OUT and v-IN in G'. In addition there are edges from n-IN to n-MID and from n-MID to n-OUT in G' for each node n in G. Show that G has a Hamiltonian circuit exactly when G' does.

Answer. Again, to show it is in NP, requires showing that testing can be performed in polynomial time. Thus, given at undirected graph G and a proposed Hamiltonian circuit, we can test it as follows: First show that the circuit is indeed a cycle in the graph. If the circuit has k nodes, then this requires k lookups in the graph G. Then check that it visits each node just once. This means that no nodes are repeated in the circuit and all nodes are included. This can be performed in k^2 operations. So overall the test is polynomial.

Now the translation from directed graphs to undirected is described above. It is clearly polynomial-time: each node and edge in the directed graph is examined just once.

Now we need to show that a Hamiltonian circuit of the undirected version G' of a directed graph graph G corresponds to a Hamiltonian circuit of the directed graph. A Hamiltonian circuit of G' passes through each i-MID just once. To do so it must reach i-IN and i-OUT, but as it is Hamiltonian, it passes through these two nodes just once. Now to reach i-IN it must have come from a j-OUT via an edge in G', but such at edge exists exactly when a directed edge from j to i exists in G. Similarly for the visit to i-IN. So, a Hamiltonian circuit in G' corresponds exactly to one in G, as required.

Test: A simpler translation from directed to undirected is simply to remove the direction from each edge in the directed graph. Show that this is not satisfactory, ie there may be Hamiltonian circuits in the undirected graph which do not correspond to Hamiltonian circuits in the directed graph. Hint: consider the directed graph $a \to b \to c$ and $a \to c$.