Homework 4
Fundamental Algorithms, Spring 2008, Professor Yap

Due: Wed Apr 2, in class.

INSTRUCTIONS:

- Please read questions carefully. When in doubt, please ask.
- Please write succinctly, to the point. Every sentence must be an complete English sentence.
- You may post general questions to the homework discussion forum in class website. Also, bring your questions to recitation on Monday.

---

1. (10 Points)
   Question 2.3 in Lect.IV (algorithm to determine if two closed paths p and q are equivalent).

   The two paths may be assumed to have the same length k. Your algorithm should run in time $O(k)$ if the path q is simple, and in time $O(k^2)$ otherwise. NOTE: you may assume the path $p = (v_0 - v_1 - \cdots - v_k)$ is represented by an array $p[0..k]$ where $p[i] = v_i$. Moreover, the vertices v_i are integers.

2. (0 Points)
   Question 3.1 in Lect.IV (reversing a graph). Do not hand in, just for your thought.

3. (3 Points)
   Question 3.4 in Lect.IV. Show that $K_{3,3}$ is nonplanar. This should be a fun problem – we will lead you through.

   HINT: assuming it is planar, then $K_{3,3}$ can be embedded in the Euclidean plane. This partitions the plane into regions (called faces). Let there be f faces. NOTE that the number f includes the infinite face (that stretches to infinity). Now, the faces are each bounded by 4 edges or 6 edges (why?). But at most one face can be bounded by 6 edges (why?). By Euler's formula, we know that $v - e + f = 2$ (note that you know $v = 6$ and $e = 9$). Let S be the set of pairs of the form (f', e') where f' is a face and e' is an edge that bounds f'. Let us compute the count $|S|$ in two ways: $s_1 := \sum_{f'} |\{(f', e') : (f', e') \in S\}|$ and $s_2 = \sum_{e'} |\{(f', e') : (f', e') \in S\}|$. Clearly, $s_1 = s_2 = |S|$. But you should get different expressions for s_1 and s_2 in terms of v, e, f, and contradict Euler's formula.

4. (12 Points)
   Question 4.8 in Lect.IV (detecting cycles in bigraphs).

5. (12 Points)
   Question 5.7 in Lect.IV (classifying edges using DFS tree of bigraphs).

6. (12 Points)
   The following is result is discussed in the notes:

   LEMMA 1. *Let u be a vertex in the DFS tree T for a bigraph G. Then u is a cut-vertex iff one of the following conditions hold:*
   (i) u is the root of T and has more than one child.
   (ii) u is not the root, but it has a child u' such that for every descendent v of u', if $v - w$ is an edge, then w is also a descendent of u. Note that a node is always a descendent of itself.

   Use this lemma to design an algorithm to detect cut-vertices in a connected bigraph. HINT: Let $ft(u)$ be the smallest value of `firstTime`$[w]$, where w is a vertex that can be reached by a back edge $v - w$, for some proper descendent v of u in the DFT tree; if there is no such back edge, then we define $ft(u)$ to be `firstTime`$[u]$. You need to address two questions: (a) How can $ft(u)$ help you determine whether a vertex v is a cut-vertex? (b) How can you compute $ft(u)$?