CMPUT 204 — Problem Set 5

(Partial solutions provided by Theo)

Topics covered in Part I are biconnected component and minimum spanning tree; in Part II are Dijkstra's algorithm for single-source shortest paths, Strassen's matrix multiplication algorithm, P & NP concepts, decision problem, and polynomial time verification.

It is highly recommended that you read pages **580**–**587**, **595**–**601**, **735**–**741**, and **966**–**983** very carefully and do **all** the exercises. The following are some of them that you are **REQUIRED** to practice on.

Quiz questions are mostly based on this list, with some minor modifications necessary. Consult your instructor and TAs if you have any problem with this list.

Part II

1. P600, Ex 24.3-1.
2. P600, Ex 24.3-2.
3. P600, Ex 24.3-4.
4. P741, Ex 28.2-3.

   Please ignore this question.
5. P741, Ex 28.2-4.

   Please ignore this question.
6. P978, Ex 34.1-1.

   Hint:

   The size of the input instance for both problems is in $\Theta(m)$, where $m \geq n$ is the number of edges in the graph.

   ''if'': Since LONGEST-PATH $\in P$, we can design an algorithm to solve LONGEST-PATH-LENGTH as follows: increase k from 0 to n and solve LONGEST-PATH until the answer is 'no'. Then $k - 1$ is the solution to LONGEST-PATH-LENGTH.

   ''only if'': Since we can solve LONGEST-PATH-LENGTH in polynomial time, we can just compare the solution to k. If k is larger, then the answer to LONGEST-PATH is 'no'; otherwise 'yes'.
7. Give a formal definition for the problem of finding the longest simple cycle in an undirected graph. Give a related decision problem.

   Hint:

   The optimization problem: Given a simple and undirected graph, compute a simple cycle that contains the maximum number of vertices (or edges).

   The decision problem:

   Instance: a simple and undirected cycle $G = (V, E)$ and an integer $k \geq 0$.

   Query: is there a simple cycle in G which contains at least k vertices?
8. P982, Ex 34.2-1.

   Hint:

   A proof or certificate to the answer 'yes' would be a 1-to-1 mapping f which maps vertices in G_1 to vertices in G_2. Therefore we design the verification algorithm to be: (1) check if the mapping is 1-to-1; (2) check if $(u, v) \in G_1$ if and only if $(f(u), f(v)) \in G_2$. This algorithm runs in $\Theta(n + m)$ time, where n and m are the number of vertices in G_1 and the number of edges in G_1.
9. P983, Ex 34.2-2.

   Hint:

   Since graph G is bipartite, every simple cycle should contain even number of edges and thus even number of vertices. A Hamiltonian cycle is a simple cycle visiting all the vertices. Therefore, if there are odd number of

vertices, then there is no Hamiltonian cycle.

10. P983, Ex 34.2-6.

    Hint:

    The proof or certificate to answer 'yes' should be a sequence of vertices. The verification process starts with checking if the two ending vertices are u and v; then checks if it is a simple path; then checks if it contains all the vertices in the given graph. The process takes time in $\Theta(n^2)$, where n is the number of vertices in G, and thus polynomial.

11. P983, Ex 34.2-7.

    Hint:

    Since the given graph is directed acyclic, we can compute a series of groups of vertices as follows:

    The group 1 contains a single vertex u;

    The group 2 contains all the vertices w such that edge $(u, w) \in G$;

    The group 3 contains all the vertices x such that edge $(w, x) \in G$, where w belongs to group 2;

    and so on up to group n.

    Since computing group $i + 1$ out of group i can be done in time $O(m)$, where m is the number of edges in G, the construction of all the n groups can be done in $O(nm)$ time. Next check if v belongs group n or not: if yes, then the answer to HAMILTONIAN-PATH is 'yes'; otherwise 'no'.

12. P983, Ex 34.2-9.

    Hint:

    Similar to the proof of ''$P \subseteq NP$'', by ignoring the proof or certificate. Check textbook page 981.