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.

Announcement: Quiz 5-1 will be on DFS and finding biconnected components.

Part I

1. P558, Prob 22-2.

   `Hint:`

   `This problem is covered by the lectures on how to compute biconnected components via DFS tree. Please check back to the lecture slides. As an example, let us prove a.`

   `Proof done using the DFS tree property: there are no cross edges.`

   `If the root of the DFS tree has degree greater than 1 (means having at least two children), then by removing root and its incident edges, there is no path connecting two vertices inside the subtrees rooted at the distinct children of root. This means the removal disconnects the graph.`

   `If the root has degree exactly 1, then removing it and the incident edges wouldn't disconnect the graph and thus the root is not an articulation vertex.`

2. P566, Ex 23.1-1.

   `Hint:`

   `Proof similar to the proof of Theorem 23.1.`

3. P566, Ex 23.1-5.

   `Hint:`

   `Proof similar to the proof of Theorem 23.1.`

4. Change the weights on the graph on page 568 by replacing each weight t with $(15 - t)$. Trace Kruskal's algorithm on the resulting graph.

5. Trace Prim's algorithm on the graph on page 571, starting with vertex d.

6. P573, Ex 23.2-2.

   `Hint:`

   `See lecture slides where the pseudocode is provided for adjacency lists representation.`

7. P573, Ex 23.2-3.

   `Please ignore this question.`

8. P573, Ex 23.2-4.

   `Hint:`

   `Assuming adjacency lists representation:`

   When all edges have weight in the range from 1 to $|V|$, the running time for sorting the edges into non-decreasing order dominates the overall running time. Therefore, the overall running time is $\Theta(|E| \log |V|)$.

   When all edges have weight in the range from 1 to a constant W, the running time for sorting the edges into non-decreasing order is $\Theta(|E|)$. Therefore the dominant term in the overall running time is the Disjoint

Sets implementation. Using operations rUnion and cFind, the overall running time is $\Theta(|E|\alpha(|V|))$.

9. P573, Ex 23.2-5.

   Hint:

   Since the dominant term in the overall running time is the heap (priority queue) implementation, the overall running times are both $\Theta(|E| \log |V|)$.

10. P573, Ex 23.2-8.

    Hint:

    The algorithm is not correct. You need to provide an example by yourself.

11. P578, Prob 23-4.

    Hint:

    (a) Yes.

    Proof can be done similarly to the proof of correctness of Prim's algorithm provided in the lecture slides.

    Can be implemented to run in $\Theta(|E|^2)$ time since checking the connectivity takes $\Theta(|E|)$ time and you might need to remove as many as $\Theta(|E|)$ edges.

    (b) No.

    (c) Yes.

    You need to figure out the implementation details by yourself.