CMPUT 204 — Problem Set 4

(Partial solutions provided by Guang)

Topics covered in Part I are dynamic programming and its applications matrix-chain multiplication and longest common subsequence; in Part II are disjoint-set and its operations, graphs, BFS and DFS.

It is highly recommended that you read pages **323–324**, **331–356**, **498–509**, and **527–549** 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. P501, Ex 21.1-1.

2. P501, Ex 21.1-3.

   `Hint:`

   `Two FIND-SET called for every edge, and thus in total` $2|E|$ `calls.`

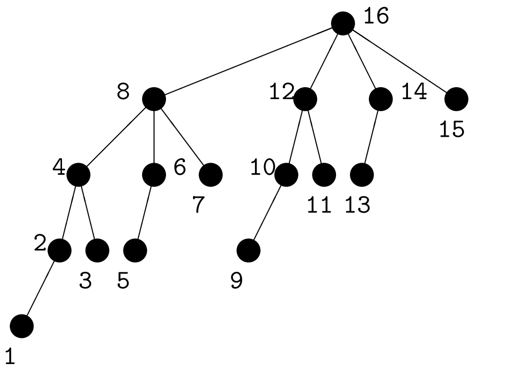
   `UNION is called when the two vertices in an edge are in different components, and it reduces the number of components by exactly` 1. `Right at the beginning there are` $|V|$ `components and the the end there are` k `components. Therefore, in total` $|V|-k$ `calls.`

3. P509, Ex 21.3-1.

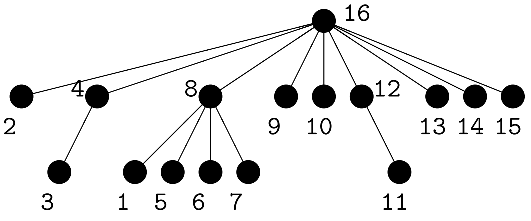
   Repeat this question without path compression.

   `Hint:`

   `By rUnion and Find, the final forest is:`



`By rUnion and cFind, the final forest is:`



4. P518, Ex 21.4-2.

   `Hint:`

   `First, show that in any subtree rooted at` x, `the rank of` x `is the height of the subtree.`

   `Second, show that the height of the subtree is at most` $\lg m$, `where` m `is the number of vertices inside the subtree.`

   `Both of them can be proved by induction. For details, see course lecture slides.`

5. P530, Ex 22.1-1.

   `Hint:`

   `For digraphs, the adjacency list for vertex` u `records the vertices` v `such that there is a directed edge from` u `to` v.

Therefore, for every vertex, its out-degree can be computed in $O(|V|)$ time (since the size of a list is at most $|V|$).

To compute the in-degree of v, we need to examine those vertices u such that (u, v) is an edge. Thus, we need to examine all other vertices and their lists, which takes $\Theta(|E|)$ time.

6. P530, Ex 22.1-2.

7. P530, Ex 22.1-3.

   Hint:

   For adjacency matrix representation (you can figure out the case for adjacency lists representation yourself):

   Examine entry (i, j), where $i \leq j$: if its value is 1 and entry (j, i) also has value 1, then their values are not changed; if its value is 1 and entry (j, i) has value 0, then entry (i, j) has a new value 0 while (j, i) has a new value 1; if its value is 0 and entry (j, i) has value 1, then entry (i, j) has a new value 1 while (j, i) has a new value 0; if its value is 0 and entry (j, i) also has value 0, then their values are not changed.

   This takes $\Theta(|V|^2)$ time.

8. P530, Ex 22.1-5.

   Hint:

   For adjacency matrix representation:

   To determine if (i, j) can have value 1, examine if there is an index k such that both (i, k) and (k, j) have value 1, which takes $O(|V|)$ time.

   Therefore, the overall algorithm will take $O(|V|^3)$ time.

9. P530, Ex 22.1-6.

   Hint:

   With the adjacency matrix representation, the idea is to avoid the examination of most of the entries by deduction:

   For every pair of vertices i and j, if entry (i, j) has value 1, then vertex i is not a universal sink; if entry (i, j) has value 0, then vertex j is not a universal sink. Therefore, start from vertex $i = 1$ to find a vertex j, if any, such that entry (i, j) has value 1. If there is no such j, then vertex i is the only candidate and thus by examining the i-th row and the i-th column we are done. Otherwise, let j be the smallest index vertex (examine the i-th row from left to right) such that (i, j) has value 1. This means vertex $k : 1 \leq k < j$ is not a universal sink. Then we can continue the examining at the j-th row, from $(j + 1)$-th column on.

   This examines in total at most $3|V|$ entries in the adjacency matrix.

10. P539, Ex 22.2-1.

11. P539, Ex 22.2-2.

12. P539, Ex 22.2-3.

    Hint:

    $\Theta(|V|^2)$.

13. P539, Ex 22.2-6.

    Hint:

    Treat every wrestler as a vertex and if two of them have a rivalry then there is an edge connecting them. Use the adjacency lists to represent the graph constructed.

    Doing a BFS on this graph and obtain the BFS tree.

    Assign the wrestlers at the odd levels to be good and those at the even levels to be bad. If there is a cross edge connecting two good guys (or two bad guys), then report that there is no possibility of designation; otherwise this is a designation of good and bad.

14. P547, Ex 22.3-2.

15. P549, Ex 22.3-10.

    Hint:

    For instance, let $v_1, v_2, \ldots, v_k$ be the vertices such that (v_i, u) is an edge;

let $w_1, w_2, \ldots, w_\ell$ be the vertices such that (u, w_j) is an edge. It could be the case that at some time during the DFS, all the w_j but none of v_i nor u have been discovered; and the DFS proceeds to discover u. In this case, a tree has to be created with its root vertex u, while the tree cannot grow since all w_j have been discovered. That is, the tree contains only u.

16. P549, Ex 22.3-11.

    Hint:

    Check the pseudocode for DFS.