CS 1501 Quiz 3 (Take-Home)

Name: Adam Karl

Pitt email address: akk67@pitt.edu

Peoplesoft number: 3919649

Fill in the fields above (name, Pitt username, and Peoplesoft number), read the instructions below and complete the honor statement, and complete your answers within this document. Then, save as a PDF for submission. Submit via Gradescope according to the instructions you received via email.

When working on this quiz, you are **allowed** to use:

* Lecture slides
* Your own notes
* The course textbook (Algorithms 4th ed. by Sedgewick and Wayne)
* Blank scrap paper

You are **not allowed** to use:

* Online resources beside those stated above
* Code or programs, written by you or others
  + Note that this means you cannot write, edit, or run code to solve these problems
* Help from a classmate, former student, or anyone else

Violating these instructions will be considered an academic integrity violation and penalized according to the course policy. Below, please confirm your agreement by adding your name to the honor statement. **Your quiz will not be graded without a completed honor statement.**

By submitting this quiz, I, Adam Karl, pledge on my honor, and with acceptance of the consequences laid out in all applicable policies, that I have neither given nor received any unauthorized assistance on this evaluation, and that the work submitted upholds the highest standards of honesty and academic integrity.

Problem 1. Consider a graph with 16 vertices (0–15). Give the contents of the id and size arrays of a Union Find data structure after performing the following operations. Use a **weighted-trees** approach with **path compression** to implement the Union Find. Resolve ties by keeping the numerically lesser connected component ID as the root.

1. union(0, 5); union(7, 10); union(9, 2); union(13, 12); union(14, 6); union(10, 15); union(4, 5); union(7, 11); union(1, 3);

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| id | 0 | 1 | 2 | 1 | 0 | 0 | 6 | 7 | 8 | 2 | 7 | 7 | 12 | 12 | 6 | 7 |
| size | 3 | 2 | 2 | 1 | 1 | 1 | 2 | 4 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 |

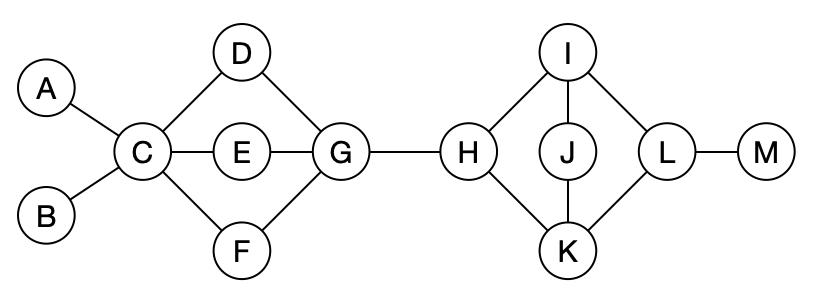
1. Starting from your answer to the (a):  
   union(6, 7); union(1, 2); union(15, 11); union(13, 4); union(3, 15);

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| id | 0 | 1 | 1 | 7 | 0 | 0 | 7 | 7 | 8 | 2 | 7 | 7 | 0 | 12 | 6 | 7 |
| size | 5 | 4 | 2 | 1 | 1 | 1 | 2 | 7 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 |

1. Starting from your answer to (b):  
   find(9); union(4, 13);

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| id | 0 | 1 | 1 | 7 | 0 | 0 | 7 | 7 | 8 | 1 | 7 | 7 | 0 | 12 | 6 | 7 |
| size | 5 | 4 | 1 | 1 | 1 | 1 | 2 | 7 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 |

Problem 2. Use an articulation point search to determine in the depicted graph is biconnected. **Start from vertex C** and assume that neighbors are seen in **alphabetical order**.



1. Give the num and low values for each vertex.

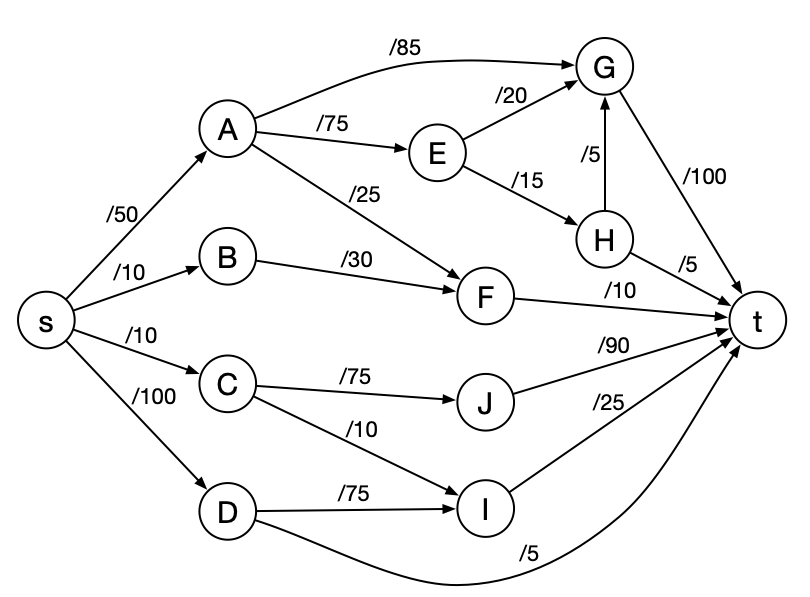
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H | I | J | K | L | M |
| num | 1 | 2 | 0 | 3 | 5 | 6 | 4 | 7 | 8 | 9 | 10 | 11 | 12 |
| low | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 7 | 7 | 7 | 7 | 8 | 12 |

1. Identify all articulation points, and for each one, concisely state how the algorithm determines it is an articulation point.

C – has >1 child (A, B, and D are all children)  
G – num of 4 <= child H’s low of 7  
H – num of 7 <= child I’s low of 7  
L – num of 11 <= child M’s low of 12

1. Finally, state if the graph is biconnected.  
     
   No (if it were, there would be no articulation points)

Problem 3. Use Edmonds-Karp to determine the max flow from s to t in the pictured graph. Assume neighbors are seen in **alphabetical order**.



1. List the augmenting paths in the order you find them. For each one, state the amount by which you increased flow along the path. (**You may not need all rows.**)

|  |  |
| --- | --- |
| Augmenting path | Flow increased |
| s D t | 5 |
| s A F t | 10 |
| s A G t | 40 |
| S C I t | 10 |
| s D I t | 15 |
| s B F A G t | 10 |
| s D I C J t | 10 |
|  |  |
|  |  |
|  |  |

1. State the max flow from s to t in the graph.  
   100
2. List the edges that comprise a min s,t-cut of the graph.  
   D-t, I-t, s-A, s-B, s-C