Marcus Anderson

Homework 4

CS 6515: Introduction to Graduate Algorithms

**Approach:**

1. To find if a path between *s* and *t* is made up of only white or gold edges within a graph, we first make a copy of the original graph, remove all gold edges, and call it *Gwhite*. We run *Gwhite* through the DFS algorithm to check if vertices *s* and *t* are connected components, if so, return TRUE. If not, we make another copy of the original graph, remove all white edges, and call it *Ggold*. We then run *Ggold* through the DFS algorithm again. If vertices *s* and *t* are connected components, return TRUE, if not, return FALSE as there is no path from *s* to *t* made up of only white edges or only gold edges.
2. To find if a path between *s* and *t* where all white edges appear before all gold edges within a graph, we first make a copy of the original graph, remove all gold edges, and call it *Gwhite*. We then run *Gwhite* through the DFS algorithm to see which vertices are connected components of *s*. Next, we make another copy of the original graph, remove all white edges from it, and call it *Ggold*. We then run *Ggold* through the DFS algorithm again, seeing which vertices are connected components of *t*. We then see if there’s a vertex reachable from *s* only using white edges and reachable from *t* only using gold edges. Return TRUE if such a vertex exists, or FALSE otherwise.

**Correctness:**

1. Running *Gwhite* and *Ggold* through the DFS algorithm will show if a path between *s* and *t* containing only white or gold edges exists if *s* and *t* are within the same connected component.
2. Running *Gwhite* and *Ggold* through the DFS algorithm will show if there’s a vertex reachable from *s* using only white edges, and from *t* using only gold edges. This vertex would indicate that a path between *s* and *t* with only white edges appearing before gold edges exists.

**Runtime:**

1. Copying the original graph takes O(n + m) time. Removing either all white or gold edges takes O(m) time. Finally, running DFS as a black box on either modified graph takes O(n + m) time. Making the overall runtime O(n + m).
2. Copying the original graph takes O(n + m) time. Removing the white and gold edges takes O(m) time. Running DFS as a black box on either modified graph takes O(n + m) time. Finally, scanning vertices takes O(n) time. Making the overall runtime O(n + m).

**Collaborators:**

* Lilley, Zachary J: zlilley3@gatech.edu
* Bertrand, James M: [jbertrand9@gatech.edu](mailto:jbertrand9@gatech.edu)
* Ramasamy, Veerajothi: vramasamy9@gatech.edu
* Acker, Joshua R: jacker7@gatech.edu
* Shah, Jeet Hemant: jshah328@gatech.edu