Andrew May
Ms Felonda

CS577 HW8

Input: A list O containing the offers that each internship makes and a list F containing the favorite internship of each student.

Output: A list S containing the students that are guaranteed to get their favorite internship.

Algorithm Analysis: First, we create a network using list O, where there is an edge of weight 1 between the source and each internship, an edge of weight 1 between the sink and each student, and an edge of weight 1 between each internship and the respective students that they offer an internship to. The reason why the edge weight between the source and each company is 1 is because each internship will only get 1 student. Similarly, the edge weight between each student and the sink is 1 because each student can only accept 1 internship. This network is comprised of a bipartite graph where the nodes on the left correspond to the internships and the nodes on the right correspond to the students. The reason why the edge weight between these nodes is 1 is because the flow can be no greater than the flow between the source and the internships.

For each edge between an internship and a student, remove the edge, and run Ford-Fulkerson on the network. If Ford-Fulkerson returns a value less than that of n, check to see if the internship and student associated with the removed edge exist as a pair in list F (i.e. the internship is the student's favorite), and if so, add the associated student to list S (i.e. this student is guaranteed to get their favorite internship) before replacing the edge and moving on to the next edge. If the internship and student associated with the removed edge do not exist as a pair in F, or if Ford-Fulkerson returns a value equal to n, then simply replace the removed egde and move onto the next one. After evaluating the network for every removed edge in the graph, return S, the list of all students guaranteed to get their favorite internship.

Andrew May
Mo Felonda

HW 8 Cont.

Correctness: Note that because both the edge weights between the source and internships, and between the students and sink, are equal to 1, then the Max-Flow of the original network should be equal to n. If the network has a max-flow less than n after removing an edge, then we can conclude that in this changed network, there is now a student without an internship and/or an internship without a student. In other words, the edge removed represents a student that was guaranteed the corresponding internship. Since we cross-examine these edges with the list of the student's favorite internships, we can then conclude if a student was guaranteed their favorite internship before adding it to the output list. Finally, since we check every edge in the bipartite graph, we can be sure that our algorithm returns a list of all the students that are guaranteed to get their favorite internship, irrespective of how the university's algorithm works

Time Complexity: There are m edges in the bipartite graph, and we run Ford-Fulkerson for each of them. The runtime of Ford-Fulkerson is $O(F \cdot (\#\text{vertices} + \#\text{edges}))$ where F is the max-flow. There are n vertices for students, n vertices for internships, 2 for source and sink, n edges between source and internships, n edges between sink and students, and m edges between internships and students. This runtime is within the parameters of the required polynomial in n and m runtime, as the runtime is $O(m(n \cdot ((2n+2) + (2n+m))))$