Junhao Zhang “Freddie”

USC ID:

CSCI 570 – HW4

08/03/2021

**1. Graded Problems**

* **[Question 1]**

First, we need to make the original graph a directed graph, by replacing each undirected edge to two directed edges each directs to the opposite direction and let *G’* denote the directed graph. Moreover, each edge of *G’* has capacity 1. Then, we are going to need sink and source nodes, but instead of connecting them to fixed nodes, we are going to do the following instead:

We iterate through all nodes, and at each iteration we let the source node connect to the current node by an edge with capacity *m*, and all the other nodes connect to the sink node by edges with capacity 1, then we get the maximum flow. The minimum among all these maximum flows is the answer.

The time complexity of this algorithm is : we need to run Ford–Fulkerson algorithm *n* times, and the time complexity of each is *O* (*Cm*) which *C* is the sum of compacities of all edges and it is also the maximum value that the maximum flow can possibly be. However, in this case since each edge has capacity 1 thus the sum of capacities is *m*, which makes the time complexity of each iteration , and we need to iterate *n* times, so the time complexity of this algorithm is .

* **[Question 2]**

We can reduce it to a maximum flow question. We would need *n* nodes and let them denote families , , … , call them “family nodes”. And need another *m* nodes and let these denote tables , , … , and call these “table nodes”. For each family node, it has edges to all table nodes and each edge has capacity 1. We will also need sink node and source node. Each table node has edge from itself to the sink node with capacity , and the source node has edges to each family node and each such edge has capacity . If the maximum flow of such network is equal to the sum of , , … , then there exists the seating assignment, otherwise doesn’t.

* **[Question 3]**

We construct a graph, called *G*, has *n* guard nodes that denote guards, and *m* interval nodes that denote each interval. For each guard can be deployed to certain time interval, then we connect the corresponding guard node to that correlate interval node by an edge with capacity 1. *G* also needs master sink node and master source node. We connect the master source node to all guard nodes, by edges with lower bound *B* and capacity *A*. We also connect all interval nodes to the master sink node, by edges with lower bound 1 and capacity 2. If such network flow has feasible solutions that can satisfy all these constraints, then there exists an expected deployment arrangement, otherwise doesn’t.

* **[Question 4]**

This can be reduced to a Maximum Flow problem. We have nodes denote clients, and we have other nodes denote base stations, we also need master sink and source nodes. First, we connect each client node to all the base nodes within the range *r* by edges with capacity 1, which will take *O* (*mn*), and *m* is number of base station nodes and *n* is number of client nodes. Then we connect the master source node to each client node by an edge with compacity 1, and we connect each base station node to the master sink node by edges with capacity *L*. And finally, we can run the Edmonds-Karp algorithm to find the maximum flow and if it is equal to *n* then there is a way to make the expected arrangement. In our graph there can be at most *mn* number of edges, thus the time complexity of Edmonds-Karp algorithm in this case is , which is polynomial.

* **[Question 5]**

This can be reduced to a Maximum Flow problem. We have nodes denote injured people, and we have other nodes denote hospitals. And we would also need master sink and source nodes. First, we connect each patient node to all the hospital nodes within half-hour’s driving time by edges with capacity 1, which will take *O* (*mn*), and *m* is number of hospital nodes and *n* is number of patient nodes. Then we connect the master source node to each patient node by an edge with compacity 1, and we connect each hospital node to the master sink node by edges with capacity . And finally, we can run the Edmonds-Karp algorithm to find the maximum flow and if it is equal to *n* then there is a way to make the expected arrangement. In our graph there can be at most *mn* number of edges, thus the time complexity of Edmonds-Karp algorithm in this case is , which is polynomial.