Computer Assignment 3

# Question 1

Suppose we have both arcs and with capacity and . When calculating the max flow, there will be a flow going through and with capacity . Hence, by subtracting from the capacity of arcs and , it will make one of the capacities become zero and the other becomes non-negative.

# Question 2

By strong law of duality, if there exists an optimal solution to the max flow problem, the solution will be the same as optimal solution in the min-cut problem. Since the min-cut problem is an s-t cut problem finding the minimum cut between the partition of s and t, in the labeling problem, all the pixels belong to partition of s will be assigned to foreground and all the pixels belong to partition of t will be assigned to background. There will be a penalty only if the neighboring pixels are in different partitions. Hence, the min-cut problem will solve the labeling problem by finding out the minimum cuts between two partitions.

# Question 3

Our method reads a network from a given file, then uses the shortest augmenting path algorithm as found in the textbook to send the maximum amount of flow on the network. Once this has completed, the residual network contains two sets of nodes – those that are reachable from the start node (), and those that are not. By using the breadth-first search reachability algorithm, we can quickly enumerate the nodes in the foreground as those reachable from , then label all other nodes as those in the foreground. This algorithm was implemented in Python version 3.6.3 on a laptop with an Intel Core i5-5200U CPU clocked at 2.20 GHz with 4 GB of RAM running Windows 10.

# Question 4

The algorithm implementation we created is correct; however, the size of the given problems is not feasible for the time constraints given, as, after nine hours of computation, the algorithm continues to find shortest augmenting paths in the network. It is unclear whether this is due to a bottleneck in the code that is underperforming, or if the programming language/machine choice themselves restrict the potential of completing these problems. However, testing on smaller examples works as expected. In the future, it may be useful to include smaller examples to test with and compare results against.

# Question 5

We will divide the arcs into four categories: the super source , the super sink , nodes which have positive values as , and nodes which have positive values as . Suppose we found a shortest path in which belongs to and belongs to . After we exhaust the path, we will drop one arc with minimum capacity. If arc is dropped, we will go back to and find a different node in to begin with. If either arc or is dropped, we will go back to directly and proceed with ’s remaining neighbors. To make it easier, we will only try ’s neighbor only if the neighbor has an arc to the super sink. To do this, we will make sure we find a path with length of at most three arcs.