ICS 311: Algorithms	Spring 2025

Problem Set 13

Kyle Berney — *Due: Friday, April 25, 2025 at 4pm*

You may discuss the problems with your classmates, however **you must write up the solutions on your own** and **list the names** of every person with whom you discussed each problem.

Start **every** problem on a separate page, with the exception that Problems 2 can start on the same page as Problem 1 (Peer credit assignment).

1 Peer Credit Assignment (1 point extra credit for replying)

Please list the names of the other members of your peer group for this week and the number of extra credit points you think they deserve for their participation in group work.

- You have a total of 60 points to allocate across all of your peers.
- You can distribute the points equally, give them all to one person, or do something in between.
- You need not allocate all the points available to you.
- ***You cannot allocate any points to yourself!*** Points allocated to yourself will not be recorded.

2 Splitting an edge in two (30 pts)

Show that splitting an edge in a flow network yields an equivalent network. More formally, suppose that flow network G contains edge (u, v), and we create a new flow network G' by creating a new vertex x and replacing (u, v) by new edges (u, x) and (x, v) with $c(u, x) = c(x, v) = c(u, v)$. Show that a maximum flow in G' has the same value as a maximum flow in G. *(If you are presenting a constructive proof, don't forget to show that the flow you construct preserves the flow conservation and the capacity constraints.)*

3 Escape Puzzle (35 pts)

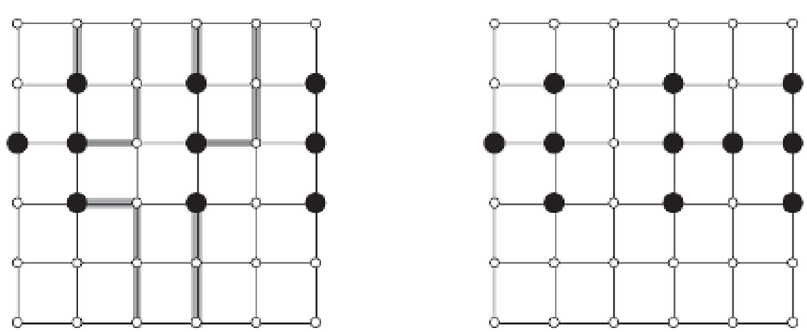

Figure 1: Examples for Problem 3: the grid in the left figure has an escape, but the grid in the right figure does not.

An $n \times n$ grid is an undirected graph consisting of n rows and n columns of vertices, as shown in Figure 1. We denote the vertex in the ith row and the jth column by (i, j). All vertices in a grid have exactly four neighbors, except for the boundary vertices, which are the vertices (i, j) for which $i = 1$, $j = 1$, $i = n$, or $j = n$.

Given $m \leq n^2$ starting points $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$, called *terminals*, in the grid, the ***escape problem*** is to determine whether or not there are m vertex-disjoint paths from the starting points to any m different points on the boundary.

Describe an efficient algorithm that utilizes network flow to solve the escape problem. Write down its pseudocode, analyze its running time and explain why it works correctly. *(Hint: remember from the class exercise how to convert a network that contains nodes with capacity constraints to one that does not.)*

4 Preventing the spread of COVID-19 (35 pts)

In 2020, the COVID-19 pandemic started spreading through Oahu. Once the rapid tests became available, to detect and contain the spread of the virus, the governor decided to install COVID-19 testing stations throughout the island. The goal was to detect any positive cases among the commuters between the Honolulu Transit Center and the Kapolei Transit Center. So he decided to place the testing stations at the road intersections and test every driver and passenger passing through that intersection. However, since the pandemic affected the state revenues from tourism, to minimize the cost (and annoyances from too many COVID tests) he wants to minimize the number of testing stations throughout the island, without allowing any commuter to travel between Honolulu and Kapolei Transit Centers bypassing testing. So he hired you to identify the minimum number of intersections where to place the testing stations. The only constrain is that due to heavy traffic, no testing stations should be placed at either the Honolulu Transit Center or at the Kapolei Transit Center.

Describe an efficient algorithm to find the minimum number of intersections to place test stations to guarantee that they intercept all traffic going between Honolulu Transit Center and Kapolei Transit Center. Write down the pseudocode, analyze its running time and prove that it works correctly. The road network is represented by an undirected graph $G = (V, E)$, with a vertex for each intersection and an edge for each road segment connecting two intersections. Two special vertices f and t represent the Honolulu Transit Center and the Kapolei Transit Center.

For example, given the graph from Figure 2 as input, your algorithm should return the number 2.

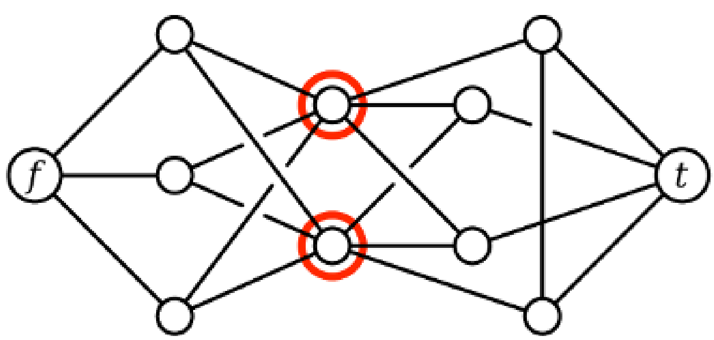

Figure 2: An example for Problem 4: The intersections where to place the testing stations are marked in red.