Saleh Alkhalifa
12/5/2022

CS 5800
Module 11

Homework #11

Problem 1. Is it a bottleneck? (10 pts)

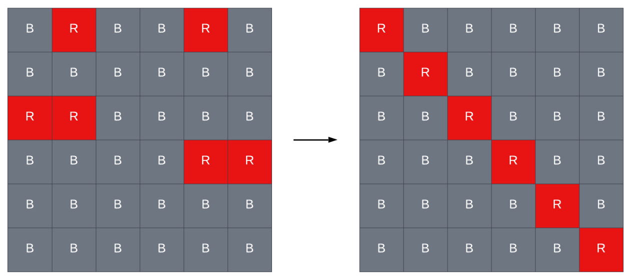
Let G=(V,E) be a flow network with source s and t sink. We say that an edge e is a bottleneck if it crosses every minimum-capacity cut separating s from t. Give an efficient algorithm to determine if a given edge e is a bottleneck in G and explain the complexity.

- We can define a flow network as a form of a directed graph G in which the edges have a capacity as well as a flow.
- In this question, we specify the graph G=(V,E) with a source s, and sink t
- We note that maximum flow can be determined for this graph via Ford-Fulkerson
    - Ford-Fulkerson:
        - Start with f(E) = 0 for each of the edges e that are a subset of E in the graph
        - While there exists a path:
            - Determine a path P from s -> t in the given residual network of graph Gf
            - You can then augment the flow along the path P
            - Update the graph
            - Finally, keep repeating this last step until complete
- We can then consider the limits of the residual graph
- If s is available from the sink, t is accessible, then a blockage edge exists
- However, if s and t are not reachable, then the process of augmenting the flow on the path from s to t cannot be done here
- Essentially, any edges that connect from the first set of nodes to the second will **be bottleneck edges**.
- For example, if we increase the capacity of a single edge we will see the final residual graph change, and yet one additional iteration of the algorithm can be done.
- We know that Ford-Fulkerson takes a running time of **O(F* |E|)**
- However, given the vertices V and edges E, we would likely see a running time of **O(|V| + |E|)** assuming we have an adjacency list representing this graph.

Problem 2. Designating marching pattern (10 pts)

The school marching band consists of students, dressed red or black, standing in an n x n grid. The music teacher wants to designate a subset of the red-dressed students such that no two red-dressed students are in the same row or column. Design and explain an algorithm that takes in a grid of red-dressed and black-dressed students and determines the size and a configuration of the largest subset of the red-dressed that can be designated.

- We know the system comprises of an n x n grid, with groups of black and red, and the objective is to reorganize this grid such that no two reds are in the same row or column



- We can begin the process by using an algorithm in an attempt to separate the two to meet the requirements above:
- First, we much check the grid both vertically and horizontally to identify the red students.
- Starting on the upper left, we can check line by line to see if there exists a red student.
- If we find one, we must check and add it to our subset. Otherwise, we continue.
- We will repeat this process until all the red students are accounted for.
- In return, we will be able to identify the size and configuration as specified above.
- Algorithm:
  - We start by instantiating a variable to monitor the size of the subset
  - Next, we create a loop to iterate over the rows, and another loop to iterate over the columns
  - For each item, we check the color of the student's clothes
    - If statement for when red is seen, we increment the count
  - We can add a red student to a new location as long as we follow the requirements of one per row and column
  - We can then repeat until there are no more red students violating the requirements above
  - Finally, we can return the size and configuration as required.
- Given the **nest nature of this loop**, we will see a time complexity of $\mathbf{O(n^2)}$

Resources:

[1] https://northeastern.instructure.com/courses/117409/pages/module-11

[2] https://en.wikipedia.org/wiki/Ford%E2%80%93Fulkerson_algorithm

[3] Introduction to Algorithms, Cormen, Third Edition. (CLRS)

[4] https://visualgo.net/en/maxflow