CS3000: Algorithms & Data — Spring 2019 — Paul Hand

Homework 5
Due Wednesday 3/13/2019 at 2:50pm via Gradescope

Name:
Collaborators:

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Wednesday 3/13/2019 at 2:50pm via Gradescope. No late assignments will be accepted. Make sure to submit something before the deadline.
- Solutions must be typeset in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly forbidden.

Problem 1. *Gerrymandering*

Gerrymandering is the practice of constructing electoral districts to lead to outcomes that favor a particular party. In this problem we will explore an algorithmic question that arises in the act of gerrymandering.

Suppose there are n *precincts* $P_1, \ldots, P_n$, where n is even, each containing M registered voters. We must divide these into two *districts*, each consisting of exactly $n/2$ precincts. For each precinct P_i, our research has shown that there are $V_{i,W}$ voters for the Whig party and $V_{i,B}$ voters for the Bull Moose Party where $M = V_{i,W} + V_{i,B}$. Our goal is to decide whether or not it is possible to gerrymander the districts so that the Whig party gets at least $\frac{Mn}{4} + 1$ votes (i.e. a majority) in each of the two districts.

For example, if we are given the following values with $n = 4, M = 100$,

Precinct	P_1	P_2	P_3	P_4
Votes for Whig Party	55	43	60	47
Votes for Bull Moose Party	45	57	40	53

then we can make one district consisting of $\{P_1, P_4\}$ and the other district consisting of $\{P_2, P_3\}$ so that the Whig party gets 102 votes in the first district and 103 votes in the second district. Thus the answer would be YES. That is, for this problem you do not need to consider how to find such a set of districts.

In this problem, you will devise a dynamic programming algorithm to output a YES or a NO for whether it is possible to assign the precincts to two districts such that the Whig party has at least $\frac{Mn}{4} + 1$ votes in each of the two districts.

(a) Describe the set of subproblems that your dynamic programming algorithm will consider. Your solution should look something like "For every [...], we define S([...]) to be [...]." Note that here your algorithm only needs to output a Boolean answer YES or NO. Here, S takes a problem instance/state (possibly with additional variables) as input, and returns YES if gerrymandering is possible and NO otherwise. You will need to determine the appropriate variables to provide to S as input.

Solution:

(b) Give a recurrence expressing the solution to each subproblem in terms of the solution to smaller subproblems.

Solution:

(c) Explain in English a valid order to fill your dynamic programming table in a "bottom-up" implementation of the recurrence.

Solution:

(d) Describe in pseudocode a ***dynamic programming*** algorithm that determines if it is possible to gerrymander the two districts so that the Whig party holds a majority of the votes in both districts. **Your implementation MUST be "bottom-up".**

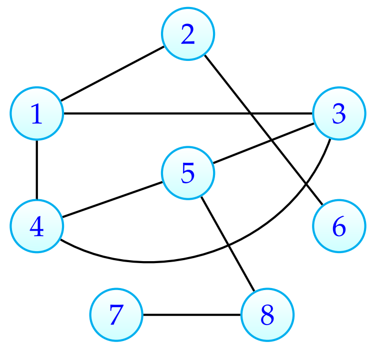
Solution:

(e) Analyze the running time and space usage of your algorithm.

Solution:

Problem 2. *Graph Representations and Exploration*

This problem is about the following undirected graph.



(a) Draw the adjacency matrix of this graph.

Solution:

Sample matrix that you need to modify.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Draw the adjacency list of this graph.

Solution:

(c) BFS this graph starting from the node 1. Always choose the lowest-numbered node next. Draw the BFS tree and label each node with its distance from 1. You could create the graph in LaTeX or include a photograph of a drawing by hand.

Solution:

Problem 3. *Graph Properties*

Consider an undirected graph $G = (V, E)$. The *degree* of a vertex v is the number of edges adjacent to v—that is, the number of edges of the form $(v, u) \in E$. Recall the standard notational convention that $n = |V|$ and $m = |E|$.

(a) Prove by induction that the sum of the degrees of the vertices is equal to $2m$.

Solution:

(b) Prove that there are an even number of vertices whose degree is odd.

Solution:

(c) Let $v \in V$ be some vertex whose degree is odd. Prove that there exists another vertex $u \in V$ such that u has odd degree and there is a path connecting v and u.

Solution: