CS3000: Algorithms & Data — Summer I '21 — Drew van der Poel

Homework 5
Due Friday, June 18 at 11:59pm via Gradescope

Name: Gabriel Peter
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 Friday, June 18 at 11:59pm 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. *No one man should have all that power* (26 points)

You work for a public relations firm and have acquired complete information on the post-election Washington DC lobbying network. In particular, you have a list V of n persons (congressmen, bureaucrats, lobbyists, businessmen, etc.) and for each person i in V, a list of persons in V that i can *influence*. (Note that it is possible that i can influence j but j cannot influence i.)

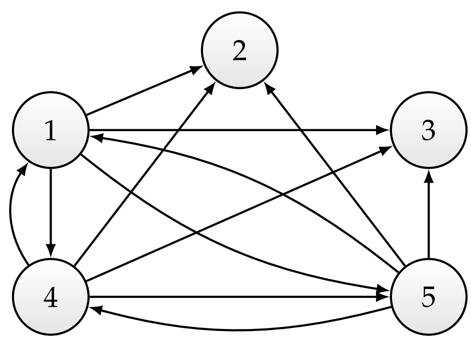
Having just studied graph algorithms, you immediately capture the above information by a directed graph G with V as the set of vertices and the set E of edges defined as follows.

$$E = \{(i, j) : i \text{ can influence } j\}.$$

Call a person i a *powerbroker* if for every other person $j \in V$, there is a path from i to j in G.

(a) **(4 points)** Give an example lobbying network with five persons in which there are exactly three powerbrokers.

Solution:



(b) Design and analyze a polynomial-time algorithm to determine *all* powerbrokers in the given lobbying network. If there are no powerbrokers, then your algorithm must indicate so.

(i) **(2 points)** Describe precisely what your algorithm is given as input and what it needs to output.[1]

Solution:

(ii) **(8 points)** Describe your algorithm. You may invoke or modify any of the graph traversal algorithms studied in class.

Solution:

(iii) **(8 points)** Justify the correctness of your algorithm. Your justification does not need to be long or formal, just convincing.

Solution:

[1] **Check:** Make sure you have this right, before you move on to designing the algorithm.

(iv) **(4 points)** Analyze the running time of your algorithm in terms of the number of vertices n and number of edges m of G.

(Any correct polynomial-time algorithm will suffice to get full credit. See if you can design an algorithm with $O(n+m)$ running time using strongly connected components.)

Solution:

Problem 2. *Stagecraft* (24+5 points)

You're in charge of assembling the stage for this year's *AlgoRhythms Music Festival (starring Master P).* Assembling the stage in time will require careful planning. You are given:

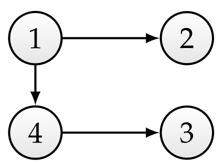
- A set V of n small tasks that are required to complete the stage.
- A set E of pairs of tasks in V. A pair (u, v) is in E if task u must be completed before task v is started. There are no cycles in E.
- For each task $u \in V$, the amount of time $t(u)$ required for the task.

You have a very large team, so you can work on any number of tasks in parallel, but you cannot start a task u until all of the prerequisite tasks v have been completed.

Design an algorithm that takes as input the graph $G = (V, E)$ (represented as an adjacency list) and the time for each task, and outputs a list consisting of the earliest possible time that each task can be completed. An example of a correct input-output is:

Input:

$t(1) = 10$
$t(2) = 11$
$t(3) = 8$
$t(4) = 5$



Output:

$c(1) = 10$
$c(2) = 21$
$c(3) = 23$
$c(4) = 15$

Your algorithm should run in $O(n + m)$ time.

(a) **(6 points)** Describe in 1-3 English sentences how you will determine the earliest possible completion times for the tasks.

Solution:

(b) **(6 points)** Describe your algorithm in pseudocode. You may make use of any algorithm we've seen in class without describing how it works, but you should clearly state what you are assuming about the algorithm.

Solution:

(c) **(8 points)** Justify that your algorithm outputs the earliest possible completion time for each project. Your justification can take any form you like as long as the argument is clear and logical.

Solution:

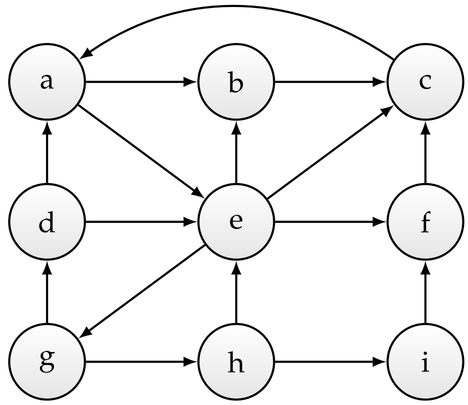
(d) **(4 points)** Analyze the running time of your algorithm. If your algorithm uses some algorithm from class as a subroutine, don't forget to include this running time in your analysis.

Solution:

(e) **(+5 points)** - In order to better your familiarity with graph algorithms, we have created a hackerrank challenge (https://www.hackerrank.com/CS3000-summer1-2021-programming-assignment-5). Please implement your algorithm and submit it to the challenge. Please list your username below. Because this problem is bonus, the course staff will generally refrain from providing assistance here.

Solution:

Problem 3. *DFS and Topological Ordering* (15 points)



Consider running depth-first search on this graph starting from node a. When there are multiple choices for the next node to visit, go in alphabetical order.

(a) **(5 points)** Label every edge as either tree, forward, backward, or cross.

Solution:

Lecture 15...

(a,b):
(a,e): tree

(b,c):

(c,a):

(d,a):
(d,e):

(e,b): tree
(e,c):
(e,f):
(e,g): tree

(f,c): tree

(g,h): tree
(g,d): tree

(h,e):
(h,i): tree

(i,f): tree

(b) **(5 points)** Give the discovery and finish times of all vertices

Solution:

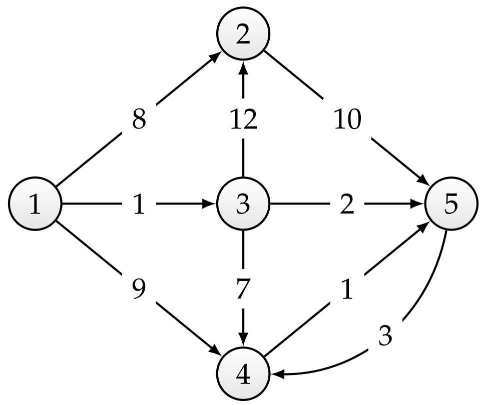
(c) **(5 points)** Is this graph a DAG? If so, give a topological ordering.

Solution:

Yes,
$[a, e, g, h, i, f, d, a, b, c]$

Problem 4. *Shortest Paths Practice* (10 points)

Use Dijkstra's algorithm the single-source shortest path problem on the following graph, using node 1 as the source node s. Write the distance from s to each node and the parent of each node in the shortest path tree in the skeleton table provided. Show your work in order to receive partial credit.



Solution:

Node	1	2	3	4	5
Distance Parent					