St. Francis Xavier University
Department of Computer Science

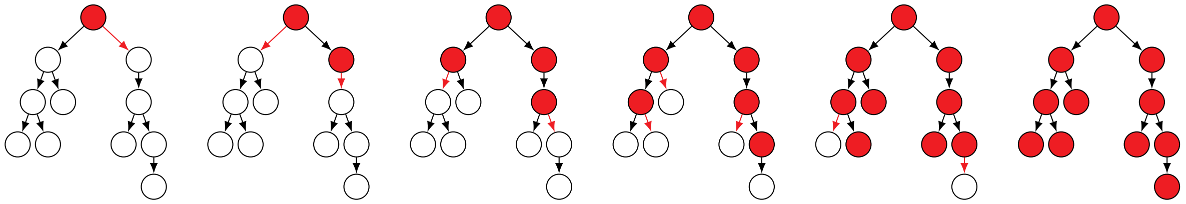
CSCI 355: Algorithm Design and Analysis
Assignment 4
Due December 4, 2025 at 11:30am

Assignment Regulations.

- This assignment must be completed individually.
- Please include your full name and email address on your submission.
- You may either handwrite or typeset your submission. If your submission is handwritten, please ensure that the handwriting is neat and legible.

[5 marks] 1. Gossip spreads fast in a small town like Antigonish, and to faciliate this spread, residents of the town have set up a *phone tree* (which is essentially a binary tree). When one designated person hears a juicy piece of news, they call up to two other people. Those people themselves call up to two other people, and the news propagates until everyone in town is aware of the story. However, a person can only be on one phone call at a time, so any person who knows the news can forward it to at most one other person per round of calls.

As an example, here is a phone tree that takes five rounds of calls before everyone hears the news:



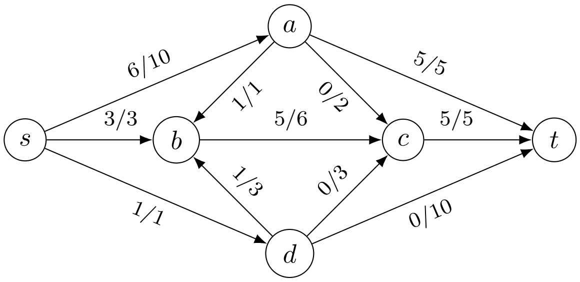
Town residents want to be able to spread gossip *even faster*, though. Given a phone tree, they would like to find out the minimum number of rounds required before all residents in the phone tree have heard the news.

(a) Let MinR$[v]$ denote the minimum number of rounds needed to spread some news from a resident v to all of the descendants of v in the phone tree. Give a Bellman equation to allow us to compute MinR$[v]$ using recurrences. Briefly explain each case of your Bellman equation.

(b) Give an algorithm to compute MinR$[r]$, where r is the root of the phone tree. What is the runtime of your algorithm?

Hint. You don't need to use any fancy techniques here.

[8 marks] 2. Consider the following flow network.



(a) What is the value of the flow shown in the network? Is this a max flow? If it is, explain why. If it is not, find a max flow.

(b) Find a min cut in the flow network, and give the capacity of your min cut.

[7 marks] 3. An *independent set* in a graph $G = (V, E)$ is a subset $S \subseteq V$ of the vertices of G such that, for any two vertices $u, v \in S$, $(u, v) \notin E$. (That is, no two vertices in S are connected by an edge.)

Suppose your benevolent professor gives you a black-box polynomial-time algorithm IndSet(G, k) that takes as input a graph G and an integer $k \geq 1$ and returns "true" if G has an independent set of size k, or "false" otherwise. Since this is a black-box algorithm, you don't know how it works: all you can do is give it the appropriate inputs and use the output it produces for you.

(a) Using the black-box algorithm IndSet(G, k), give an algorithm that outputs the size of the *largest* independent set in a given graph G. Your algorithm must run in polynomial time.

(b) Using the black-box algorithm IndSet(G, k) as well as your algorithm from part (a), give an algorithm that outputs an independent set in a given graph G that has maximum size. Your algorithm must run in polynomial time.

Note. The difference between this question and the previous question is that you must now output the set, and not just the size of the set.

(c) We know that the independent set decision problem is NP-complete. What does this imply about the black-box algorithm IndSet(G, k) given to you by your benevolent professor, and about the algorithms you gave in parts (a) and (b)?

[5 marks] 4. Choose your favourite topic from the course, and write a multiple-choice style question with one correct answer and 3–4 plausible-but-incorrect answers that tests a concept or notion related to that topic.

For inspiration, consider the multiple-choice style questions you saw on the midterm and practice midterm exams.