

Analysis of Algorithms
CS 375, Spring 2021

Problem Set: Graphs

Due **BY THE BEGINNING OF CLASS** Wednesday, April 21

- From your textbook (CLRS), please read Chapters 22.1–22.3.
- To submit this Problem Set, the standard file naming and HW submission conventions for CS375 apply: Please put your answers in a PDF file named

CS375_PS_Graphs_<userid>.pdf

where <userid> is replaced by your full Colby userid (including your class year)—for example, my file would be called `CS375_PS_Graphs_eaaron.pdf`—and submit it to your `SubmittedWork` folder.

- Unless otherwise specified, exercises will be from the CLRS textbook and will be named on HW assignments by exercise number used in the book.
- **IMPORTANT:** Some of these exercises may build upon topics covered in class meetings after the day this is assigned; they are included here “early” so you can see all of the exercises on this assignment.
- Please recall the guidelines for algorithms given on previous HW assignment sheets. They continue to apply to all HWs in CS375.
- In general, there may be multiple correct ways of presenting an algorithm, although excessively inefficient or inelegant solutions may not receive full credit. If you have questions about whether your proposed solution is excessively inefficient or inelegant, please ask your Prof.!
- *A general note for CS375:* As always, please present answers cleanly and **explain them clearly and thoroughly**, giving all details needed to make your answers easy to understand; typed-up (rather than handwritten) answers are especially appreciated. (Feel free to talk with your Prof. or TA’s about using L^AT_EX to typeset your answers!) Graders may not award full credit to incomplete or illegible solutions. Clear communication *is* the point, on every assignment.

In general in CS375, unless explicitly specified otherwise, answers should be accompanied by explanations. Answers without explanations may not receive full credit. Please feel free to ask me any questions about explanations that might come up!

Exercises

1. CLRS 22.1-2 (pg. 592). A lengthy explanation is not necessary; it is sufficient to present a matrix, a list of lists, and enough detail to make sure a reader can understand how they correctly represent the tree. **Note:** A *complete binary tree* is one in which every level is filled in, left to right; for example, a complete binary tree with 7 vertices has every possible node on three levels. As a default—and for this exercise in particular—edges in trees are considered to be undirected. (See pg. 1179 for a definition of a complete tree.)

2. CLRS 22.2-2 (pg. 601). Give your answer by presenting a table of the following form, with the values for discovery time and predecessor filled in for each vertex; no further explanation is necessary for this exercise.

vertex	discovery time (field .d)	π
r		
s		
t		
u		
v		
w		
x		
y		

3. Show how depth-first search works on the graph of Figure 22.6 in CLRS by showing the resulting discovery time, finishing time, and π values. Similar to the above exercise, give your answer by presenting a table of the following form; no further explanation is needed for this exercise.

vertex	discovery time (field .d)	finishing time (field .f)	π
q			
r			
s			
t			
u			
v			
w			
x			
y			
z			

Assume that the **for** loop of lines 5–7 of the DFS procedure considers the vertices in alphabetical order. Furthermore, assume that the graph has an adjacency list representation, and assume that each adjacency list is ordered alphabetically.