Graphs

Questions

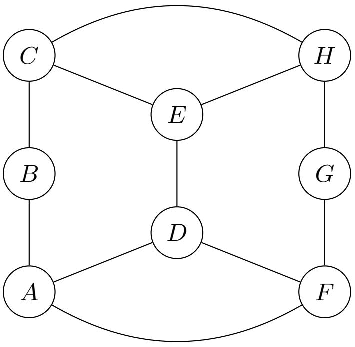
Problem 1. Draw a graph $G = (V, E)$ given by:

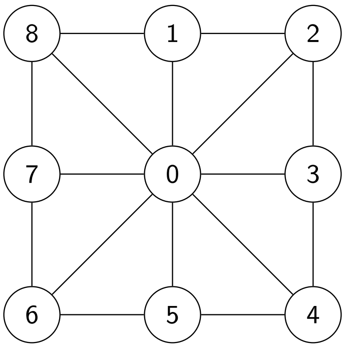
- $V = \{1, 2, 3, 4, 5, 6\}$
- $E = \{\{1, 2\}, \{1, 4\}, \{3, 2\}, \{4, 5\}, \{5, 1\}, \{5, 2\}\}$

Is it connected? If not, how many connected components does it have?

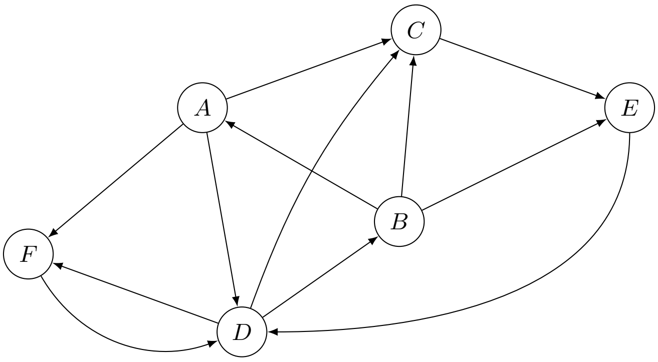
Problem 2. Consider an undirected graph with five vertices, all of them with degree 3. If such a graph exists, draw it. If it doesn't, give a reasoned explanation.

Problem 3. Show how the following pair of graphs are represented using an adjacency matrix and adjacency lists, and count the degree of each vertex.





Problem 4. List all the possible visit sequences of the vertices of the graph below, applying a depth-first traversal that starts at vertex E. Break all ties by picking the vertices in alphabetic order.



Problem 5. Harray Plotter and her three wizard friends have been tasked with searching every room of a Labyrinth for magical artifacts. The Labyrinth consists of n rooms, where each room has at most four doors leading to other rooms. Assume all doors begin closed and every room in the Labyrinth is reachable from a specified entry room by traversing doors between rooms. Some doors are protected by evil enchantments that must be disenchanted before they can be opened; but all other doors may be opened freely. Given a map of the Labyrinth marking each door as enchanted or not, describe an $O(n)$-time algorithm to determine the minimum number of doors that must be disenchanted in order to visit every room of the Labyrinth, beginning from the entry room.

Problem 6. ABC is a cloud computing company where users can upload *computing jobs* to be executed remotely. The company has a large number of identical cloud computers available to run code, many more than the number of pieces of code in any single job. Any piece of code may be run on any available computer at any time. Each computing job consists of a code list and a dependency list.

A *code list* C is an array of code pairs $(f, t) \in C$, where string f is the file name of a piece of code, and t is the positive integer number of microseconds needed for that code to complete when assigned to a cloud computer. Assume file names are short and can be read in $O(1)$ time.

A *dependency list* D is an array of dependency pairs $(f1, f2) \in D$, where f_1 and f_2 are distinct file names that appear in C. A dependency pair (f_1, f_2) indicates that the piece of code named f_1 must be completed before the piece of code named f_2 can begin. Assume that every file name exists in some dependency pair.

(a) A job (C, D) can be *completed* if every piece of code in C can be completed while respecting the dependencies in D. Given job (C, D), describe an $O(|D|)$-time algorithm to decide whether the job can be completed.

(b) ABC wants to know how fast they can complete a given job. Given a job (C, D), describe an $O(|D|)$-time algorithm to determine the minimum number of microseconds that would be needed to complete the job (or return that the job cannot be completed).

References

Atserias, Albert et al. (2022). *Data Structure and Algorithms Problem Set*. Universitat Politécnica de Catalunya.