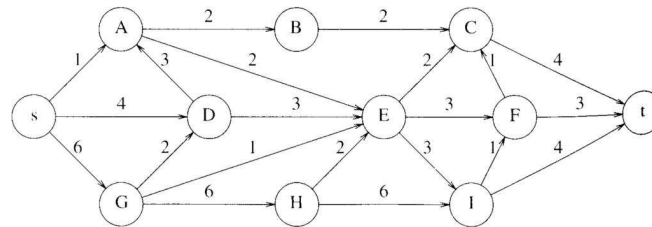


9.1 Find a topological ordering of the graph:



Ans: The following ordering is arrived at by using a queue and assumes that vertices appear on an adjacency list alphabetically. The topological order that results is then

$s, G, D, H, A, B, E, I, F, C, t$

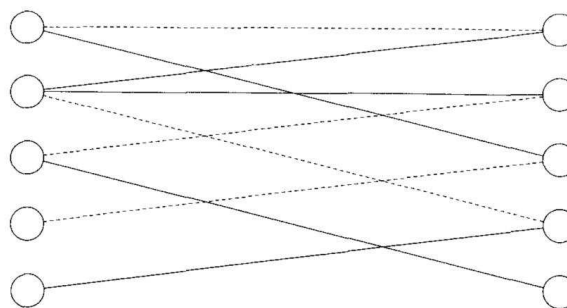
9.2 If a stack is used instead of a queue for the topological sort algorithm, does a different ordering result? Why might one data structure give a “better” answer?

Ans: Assuming the same adjacency list, the topological order produced when a stack is used is

$s, G, H, D, A, E, I, F, B, C, t$

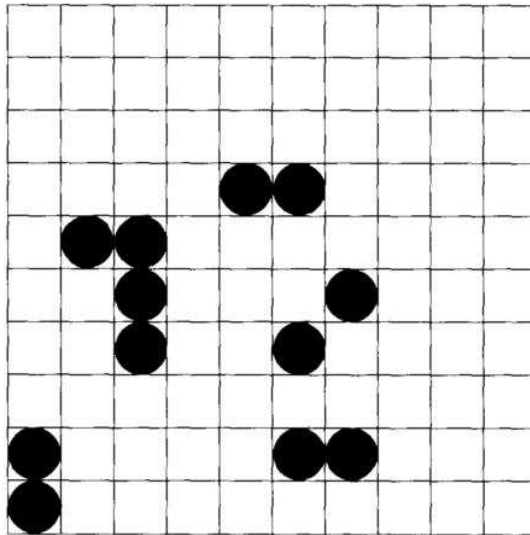
Because a topological sort processes vertices in the same manner as a breadth-first search, it tends to produce a more natural ordering.

9.13 A bipartite graph,  $G = (V, E)$ , is a graph such that  $V$  can be partitioned into two subsets  $V_1$  and  $V_2$  and no edge has both its vertices in the same subset. See the figure below.



- Show that every tree is a bipartite graph.
- Give a linear algorithm to determine whether a graph is bipartite.
- The bipartite matching problem is to find the largest subset  $E'$  of  $E$  such that no vertex is included in more than one edge. A matching of four edges (indicated by dashed edges) as shown in the figure above. There is a matching of five edges, which is maximum.  
Show how the bipartite matching problem can be used to solve the following problem: We have a set of instructors, a set of courses, and a list of courses that each instructor is qualified to teach. If no instructor is required to teach more than one course, and only one instructor may teach a given course, what is the maximum number of courses that can be offered?

- Ans:* (b) Assume that the graph is connected and undirected. If it is not connected, then apply the algorithm to the connected components. Initially, mark all vertices as unknown. Pick any vertex  $v$ , color it red, and perform a depth-first search. When a node is first encountered, color it blue if the DFS has just come from a red node, and red otherwise. If at any point, the depth-first search encounters an edge between two identical colors, then the graph is not bipartite; otherwise, it is. A breadth-first search (that is, using a queue) also works. This problem, which is essentially two-coloring a graph, is clearly solvable in linear time. This contrasts with three-coloring, which is NP-complete.
- (c) Construct an undirected graph with a vertex for each instructor, a vertex for each course, and an edge between  $(v, w)$  if instructor  $v$  is qualified to teach course  $w$ . Such a graph is bipartite; a matching of  $M$  edges means that  $M$  courses can be covered simultaneously.
- 9.45** Consider an  $N \times N$  grid in which some squares are occupied by black circles. Two squares belong to the same group if they share a common edge. In the figure below, there is one group of four occupied squares, three groups of two occupied squares, and two individual occupied squares. Assume that the grid is represented by a two-dimensional array. Outline an algorithm in 1–2 sentences that does the following:
- Computes the size of a group when a square in the group is given.
  - Compute the number of different groups.
  - List all groups.



*Ans:* These are all recursive depth first searches!

- 9.51** The input is a collection of currencies and their exchange rates. Is there a sequence of exchanges that makes money instantly? For instance, if the currencies are  $X$ ,  $Y$ , and  $Z$  and the exchange rate is  $1X$  equals  $2Y$ s,  $1Y$  equals  $2Z$ s, and  $1X$  equals  $3Z$ s, then  $300Z$ s will buy  $100X$ s, which in turn will buy  $200Y$ s, which in turn will buy  $400Z$ s. We have thus made a profit of 33 percent. State the general idea and you are not required to even outline the algorithm steps.

*Ans:* Each currency is a vertex; draw an edge of cost  $\log C$  between vertices to represent a currency exchange rate,  $C$ . A negative cycle represents an arbitrage play.