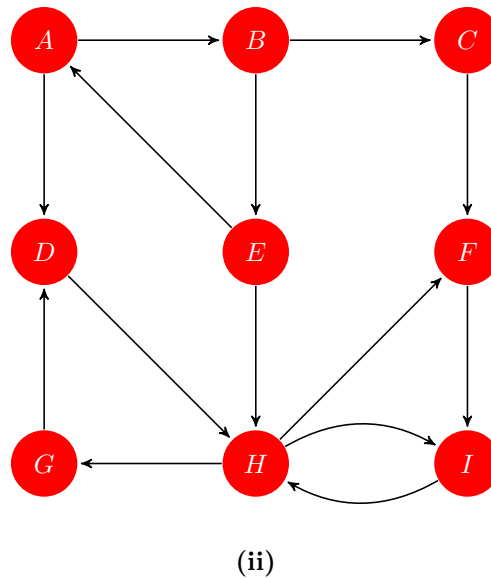


Exercises

Dasgupta 3.4 a-c (apply only to graph ii)

Run the strongly connected components algorithm on the following directed graphs G . When doing DFS on G^R : whenever there is a choice of vertices to explore, always pick the one that is alphabetically first.



In each case answer the following questions.

- In what order are the strongly connected components (SCCs) found?
- Which are source SCCs and which are sink SCCs?
- Draw the "metagraph" (each meta-node is an SCC of G).

Dasgupta 3.13 a-c (note that a asks for a formal proof)

Undirected vs. directed connectivity.

- Prove that in any connected undirected graph $G = (V, E)$ there is a vertex $v \in V$ whose removal leaves G connected. (Hint: Consider the DFS search tree for G .)
- Give an example of a strongly connected directed graph $G = (V, E)$ such that, for every $v \in V$, removing v from G leaves a directed graph that is not strongly connected.
- In an undirected graph with 2 connected components it is always possible to make the graph connected by adding only one edge. Give an example of a directed graph with two strongly connected components such that no addition of one edge can make the graph strongly connected.

Dasgupta 3.16

Suppose a CS curriculum consists of n courses, all of them mandatory. The prerequisite graph G has a node for each course, and an edge from course v to course w if and only if v is a prerequisite for w . Find an algorithm that works directly with this graph representation, and computes the minimum number of semesters necessary to complete the curriculum (assume that a student can take any number of courses in one semester). The running time of your algorithm should be linear.