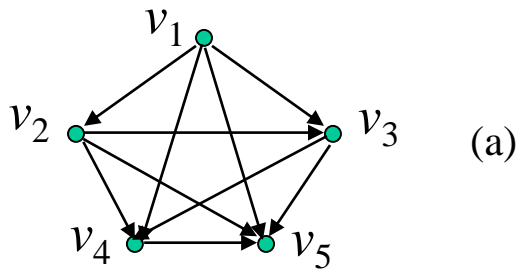


Assignment #2

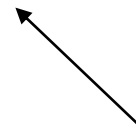
(Assignment due: Nov. 06, 2024)

- 1.(30) See the graph G shown in Fig. 1(a), which can be stored in a file on disk as a set of pairs as shown in Fig. 1(b). Design an algorithm to load the file in main memory and store it as a set of linked lists. (The algorithm should be able to be used for any graph.)



(v₁, v₂)
(v₁, v₃)
(v₁, v₄)
(v₁, v₅)
(v₂, v₃)
(v₂, v₄)
(v₂, v₅)
(v₃, v₄)
(v₃, v₅)
(v₄, v₅)

(b)



Each pair represents an edge.

Fig. 1

Assignment #2

2.(30) Another way to perform topological sorting on a directed acyclic graph $G = (V, E)$ is to repeatedly find a node of in-degree 0, output it, and remove it and all of its outgoing edges from the graph. Explain how to implement this idea so that it runs in time $O(|V| + |E|)$.

3.(40) Consider the graph G shown in Fig. 1(a) once again. Its DFS tree T is shown in Fig. 2(a), in which each node v is associated with $(d[v], f[v])$. Remember that a node is a descendant of another node v in T if $d[v] < d[u] < f[u] < f[v]$.

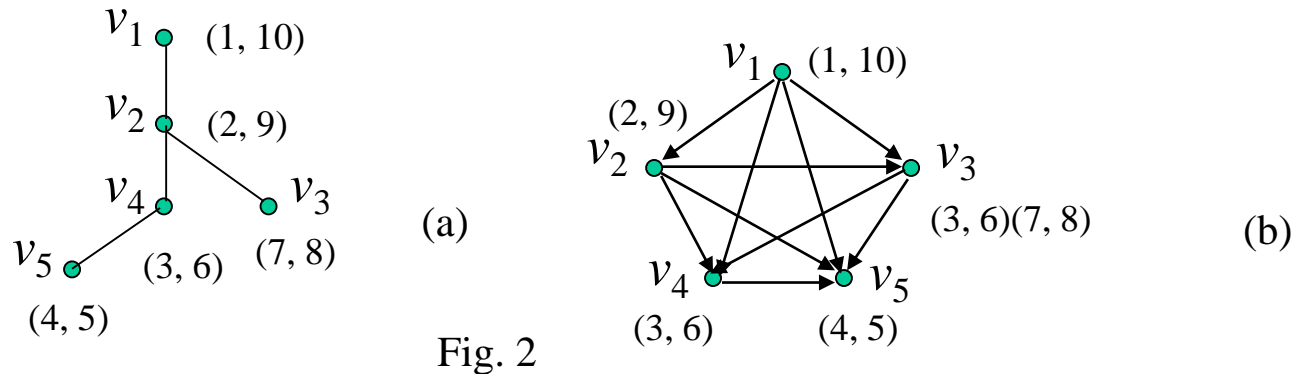


Fig. 2

Describe an algorithm that associates each node v in G with a sequence s of pairs (as shown in Fig. 2(b)) such that u is a descendant of v in G if and only if

$$d[v] < d[u] < f[u] < f[v], \text{ or}$$

there exists a pair $p = (a, b)$ in s with $a \leq d[u] < f[u] \leq b$.