1. Draw the complete undirected graphs on five vertices. (5%)

2. For the digraph G shown as follows.

(1) Write out in-degree and out-degree of the vertices 1, 3, 5. (6%)
(2) Write out the strongly connected components of G. (4%)
(3) Write out the adjacency matrix representation of G. (4%)
(4) Write out the adjacency list representation of G. (4%)
(5) Find a cycle in subgraph G

3. Given a graph G as below.

(1) Write out the DFS and BFS sequences, respectively.
   (由 A 開始，並以字母順序為主) (6 %)
(2) Show the articulation points in this graph. (2 %)
(3) Show the biconnected components. (6%)
(4) Find minimum-cost spanning tree. (6 %)
   (請寫出你使用的演算法名稱)

4. Given a Min-Max Heap as below.

(1) Insert an element 7.  (4%)
(2) Insert an element 21.  (4%)
(3) Delete a min element. (4%)

5. Given an AOV (activity-on-vertex) graph as follows.

(1) Write out a topological order. (4 %)
(2) Write out an algorithm that can find topological orders. (4 %)

試題務請以黑色筆書寫，俾便製版印刷。
6. 請回答有關 hash table 的問題
   (1) 何謂 uniform hashing function? 4%
   (2) 請舉出四種常用的 uniform hashing function。4%
   (3) 何謂 collision? 何謂 overflow? 何時 collision 與 overflow 會同時發生? 6%
   (4) 有哪二種方法可以用來處理 overflow 的情形? 理論上哪一種方法的效能較佳? 6%
   (5) 有一些 identifier 分別為 GA, D, A, G, L, A2, A1, A3, A4，若我們以這些 identifier 的第一個字母順序當作 hash function，處理 overflow 機制則分別以(4)中的二種方法進行，請圖示最後結果。10%

7. Given a Deap as below.
   (1) the deap after inserting two new elements
      with keys 5 and 80, respectively. 5%
      (the node with key 5 is inserted first)
   (2) the deap after deleting two min elements
      from the above 11-element deap. 5%

8. Given an AOE (activity-on-edge) graph as above, please show
   (1) The earliest time of events 4 and 7. 2%
   (2) The earliest time of activities a_4 and a_7 2%
   (3) The latest time of activities a_4 and a_7 2%
   (4) Is a_5 a critical activity? Why? 4%

9. Given a Graph G as follows, please find shortest path from vertex 0 to the other vertices.
   (1) Show the length-adjacency list representation of G. 5%
   (2) Show the action of algorithm Shortest Path that
      given by Edsger Dijkstra step by step. 10%