1. (A) Explain what an ADT(Abstract Data Type) is. (5%)
   (B) Describe an ADT of the stack data structure.(5%)
   (C) Implement the stack ADT which you described in (B) using two queues. (You may use a pseudo algorithm language or any programming language.)(10%)

2. For any non-empty binary tree T, let n denote the total number of nodes in T. The number of leaf nodes in T is denoted as L. Let n_1 and n_2 denote the number of degree-1 nodes and degree-2 nodes respectively. For the types of T listed below, which expression(s) is(are) true? (Note: multiple choices are possible.)

   (A) Strict binary tree. (5%)   (a) $n = 2L - 1$
   (B) Arbitrary binary tree. (5%)   (b) $n_2 = L - 1$
   (C) AVL tree.(5%)   (c) $n_1 + 2n_2 = n - 1$
   (D) Complete binary tree. (5%)   (d) $(n_1 + n_2) \le L \le (n_1 + n_2 + 1)$
   Please choose from right: (e) None of the above.

3. Given the adjacency matrix representation of a graph G as

   $$M_G = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

   Nodes in G are denoted by integer 0, 1, 2, ...
   (A) Draw the BFS(1) spanning tree of G. (6%)
   (B) List the articulation points of G. (4%)
   (C) Explain the meaning of a_{ij} in $M_G \times M_G$ (ordinary matrix multiplication)(5%)

4. The worst case time complexity is $O(n\log n)$ for heap sort and is $O(n^2)$ for quick sort, while their average case time complexities are both $O(n\log n)$. Why people prefer quick sort anyway ? (15%)

5. Show that if T is a spanning tree for the undirected graph G, then the addition of an edge e, $e \notin E(T)$ and $e \in E(G)$, to T creates a unique cycle. (15%)

6. Write a recursive procedure, in pseudo language, for computing the binomial coefficient $\binom{n}{m} = \dfrac{n!}{m!(n-m)!}$, where $\binom{n}{0} = \binom{n}{n} = 1$. Analyze the time and space requirements of your algorithm. (15%)