1. (15%) The first pass of quick sort is to partition the input data into two sets of data, one contains all input data that are smaller than a pivot value and the other set contains all input data that are greater than or equal to the pivot value. Under what condition will the quick sort run in $O(n^2)$ in the worst case ? And under what condition will the quick sort run in $O(n\log n)$ in the worst case ? And how will you change your quick sort algorithm to make it always run in $O(n\log n)$ in the worst case ?

2. (15%) Heap is an example of implementing a binary tree without pointers. Thus, heap uses the so-called *implicit data structure*, i.e., uses the address and/or the contents of a node to decide the addresses of its children. Explain the way you use to decide the addresses of two children for a heap node. Also give (design) at least another two examples of implicit data structure to implement linked list or binary tree without explicit pointers.

3. (20%) Answer the following questions.
   (a) What data structure do you need when you try to translate a recursive program into an iterative program?
   (b) What is the height of a balanced binary tree with $n!$ leaves? Give your answer in big O notation.
   (c) Under what condition will you use linked list, instead of an array, to implement a stack?
   (d) Give an example of the real world that you have to find out the minimal spanning tree of an undirected weighted general graph.

4. (5%) Let $f(n) = \sum_{i=1}^{n} \log i$, show that $f(n) = O(n \cdot \log n)$.

5. (8%) Let *BFS(i)* and *DFS(i)* denote the outcomes of visiting all nodes in a graph G starting from node i by *breadth-first search* and *depth-first search* respectively. Given a directed graph $G = (V, E)$ and $V(G) = \{A, B, C, D, E\}$. Please answer the following questions:
   (a) Suppose $BFS(A) = DFS(A)$, draw one possible configuration of the graph G. (4%)
   (b) Suppose $BFS(A) = DFS(A)$, $BFS(C) = DFS(C)$, and G is connected but G is not cyclic. draw one possible configuration of the graph G. (4%)

6. (4%) Draw the binary tree which assumes the outcomes of in-order and post-order traversal:
   In-order: A B C D E F G H J K
   Post-order: A C E D B J H K G F

7. (18%) Build an AVL tree by inserting the following character strings of in the given order:
   DUS, JFK, ZRH, HKG, KHH, FRA, ARN, LBA, MEX, GLA, ORY, MAN, TPE, ORD, NAP.
   (a) Describer your comparator function. (3%)
   (b) Draw the AVL tree. (6%)
   Insert a string **NYC** into the AVL tree you obtained in *(b)*
   (c) What kind of rotation (e.g., LL, RR, LR, RL) is required? (3%)
   (d) Sketch the rotation of keeping the tree balanced. (6%)

8. (15%) Explain the following terms:

(a) 2-3 tree	(3%)	*(b)* ADT	(3%)
(c) Biconnected graph	(3%)	*(d)* Hash function	(3%)
(e) Postfix notation	(3%)		