考試科目	資料結構	系所別	資訊管理學系/科技組	考試時間	2 月 7 日(五) 第 四 節

Answer the following questions:

1. Find the longest common subsequence of X and Y, where

   X = BAACBBSCHAA
   Y = ABABCSCAH

   1.1 (6%) Let L[i ,j] denote the longest common subsequence of X[0..i] and Y[0..j]. (X[0..-1] and Y[0..-1] denote an empty string.). Show the equivalence relation on L[i,j] with L[i-1, j-1], L[i-1, j] and L[I, j-1] and the base cases.
   1.2 (10%) Describe a dynamic programming algorithm with the above equivalence relation to find the longest common subsequence between two strings X[0..m] and Y[0..n].
   1.3 (10%) Run the algorithm on the above X and Y and show the complete table on L[i, j] (-1<=i<=10, -1<=j<=8) to derive the value of L[10, 8].

2. AVL Binary Search Trees:
   2.1 (10%) Build an AVL binary search tree by inserting the following keys. (Hint: rebalance tri-nodes when the difference of heights of sub trees is larger than 1)
   5, 6, 12, 7, 11, 35, 28, 16, 13, 29, 20, 2, 58, 18, 22, 42.
   2.2 (5%) From 2.1, show the tree after removing the key 22
   2.3 (5%) From 2.2, show the tree after removing the key 35

3. Hash Tables:
   Consider a hash table storing the following keys:
   98, 77, 19, 29, 41, 82, 12, 108, 53, 25, 54, 42.
   Let N=17. h(k) = k mod 17.
   3.1 (8%) Show the hash table that handles collision with separate chaining.
   3.2 (8%) Show the hash table that handles collision with linear probing.
   3.2 (8%) Show the hash table that handles collision with double hashing. (Let d(k) = 11- k mod 11.)

4. Graph Constructions:
Tom plans to travel around Taiwan. Below is the cost between two places (undirected).

   Taipei, Kaoshiung, 850

備註	一、作答於試題上者，不予計分。 二、試題請隨卷繳交。

考試科目	資料結構	系所別	資訊管理學系/科技組	考試時間	2 月 7 日(五) 第 四 節

Taipei, Taidong, 750
Taipei, Hualian, 500
Hualian, Taidong, 350
Hualian, GreenIsland, 1650
Taidong, Kaoshiung, 450
Taidong, GreenIsland, 800
Kaoshiung, Kenting, 400
Taidong, Kenting, 550
Taipei, Taichung, 400
Taichung, Nanto, 200
Nanto, Hualian, 700

4.1 (10%) Draw an edge-list structure to represent the graph. A node carries the name and an edge carries the edge.
4.2 (10%) Define a DFS algorithm to traverse all the nodes. Run the algorithm starting from "Taipei" and show the nodes and edges that are visited.
4.3 (10%) Define a BFS algorithm to traverse all the nodes. Run the algorithm starting from "Kaoshiung" and show the nodes and edges that are visited.