參考用

1. There are N courses in a computer science department. Each course contains two data fields: (1) number (3 digits) as key and (2) name (string). Insert the following 4 courses:

123 Data Structure

234 Software Engineering

345 Programming Language

233 Compiler

into each of the 5 data structures below, and draw a sketch for each. Note that you should draw each and every data field, including pointers. Null pointers should be specified as ●

(1) array (5%)

(2) doubly-linked list (need a dummy node) (5%)

(3) red-black tree (need a color field) (5%)

(4) chained hashing with a hash table of size 11 and a hash function h where h(key) = key mod 11 (5%)

(5) heap (5%)

2. Insert the integers 4, 2, 6, 1, 3, 5, 7 into an initially empty binary search tree.
   (1) Show the resulting tree. (4%)
   (2) Show the result of deleting 4 and 2 from the tree in (1) above (4%)

3. The following array represents a binary tree. The first row is the index, and the second row indicates the content of each node. The concept here is similar to the implementation of (min or max) heap using the array.

注意：背面有試題

參考用

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	B	C		D	E						F	G							

(1) Is it possible for index 14 to contain some information? Why or Why not? (2%)

(2) Give the result of the inorder traversal of the tree. (2%)

(3) Give the result of the preorder traversal of the tree. (2%)

(4) Give the result of the postorder traversal of the tree. (2%)

(5) Give the result of the level order traversal of the tree. (2%)

4. Consider the following variation of the Minimum Spanning Tree Problem: Given a graph G of n vertices and m edges AND a minimum spanning tree T of graph G, we wish to add a new edge e with weight w_e to G forming a new graph G' and construct the minimum spanning tree of the new graph G'.

   (1) Give an algorithm which constructs the minimum spanning tree of G' in time faster than Prim's or Kruskal's algorithm. (Zero credit for giving an algorithm which simply constructs the minimum spanning tree of G' from scratch. For full credit, your algorithm should run in $O(n)$ time.) (12%)

   (2) Prove (argue convincingly) your algorithm is correct. (8%)

   (3) Analyze the running time of your algorithm. (4%)

5. For an undirected graph $G=(V, E)$, a vertex v in V, and an edge (x,y) in E, let $G\backslash v$ denote the subgraph of G obtained by removing v and all the edges incident to v from G; and let $G\backslash(x,y)$ denote the subgraph of G obtained by removing the edge (x,y) from G. If G is connected, then $G\backslash v$ can be disconnected or connected. For

example, if G is a tree, $G\backslash v$ is connected for any leaf node v; otherwise, it is disconnected. Note also that if G is a tree, $G\backslash(x,y)$ is disconnected for any edge (x,y).

(1) Prove that for any connected graph G, we can always find a vertex v in G such that $G\backslash v$ is connected. (5%)

(2) Given a connected graph G, design an $O(|V|)$ time algorithm to find a vertex v in G such that $G\backslash v$ is connected. (8%)

(3) Given a connected graph G, design an $O(|V|)$ time algorithm to either find an edge (x,y) in G such that $G\backslash(x,y)$ is connected or report that no such an edge exists. (8%)

6. On the real line, a closed interval $[a, b]$ is said to be ***unit-length*** if $b - a = 1$. Given a set $\{x_1, x_2, \ldots, x_n\}$of n points on the real line, we say that a set S of unit-length closed intervals ***covers*** $\{x_1, x_2, \ldots, x_n\}$ if each point x_i is contained in some interval of S. The problem is to find such an S with minimal size $|S|$. Assume that the given points are sorted as $x_1 \leq x_2 \leq \ldots \leq x_n$. Design an $O(n)$ time algorithm to solve this problem. (12%)