國立臺灣海洋大學一〇二學年度研究所碩士班暨碩士在職專班招生考試試題

考試科目： 基礎計算機科學（含資料結構、演算法）

系所名稱： 資訊工程學系碩士班不分組

1.答案以橫式由左至右書寫。2.請依題號順序作答。

---

1. (15%) Given a set S of n integers and another integer x, write a $\Theta(n\lg n)$-time algorithm to determine whether or not there exist two elements in S whose sum is exactly x. (Hint: Sorting)

2. (10%) Describe the **Floyd-Warshall algorithm** that, given an $n \times n$ matrix W representing the edge weights of an n-vertex directed graph $G = (V, E)$, solves the all-pairs shortest-paths problem in $\Theta(n^3)$ time.

3. (15%) Let $G = (V, E)$ be a connected, undirected graph with edge-weight function $w : E \to \mathbf{R}$, and assume all edge weights are distinct. Consider a cycle $\langle v_1, v_2, \ldots, v_k, v_{k+1} \rangle$ in G, where $v_{k+1} = v_1$, and let (v_i, v_{i+1}) be the edge in the cycle with the largest edge weight. Prove that (v_i, v_{i+1}) does **not** belong to the minimum spanning tree T of G.

4. (10%) Analysis the time complexity of the BUILD-MAX-HEAP procedure.
(Hint: For an n-element heap, height is $\lfloor \lg n \rfloor$ and at most $\lceil n/2^{h+1} \rceil$ nodes of any height h.)

```
MAX-HEAPIFY(A, i)
1.  ℓ ← LEFT(i)
2.  r ← RIGHT(i)
3.  if ℓ ≤ heap-size[A] and A[ℓ] > A[i]
4.      then largest ← ℓ
5.      else largest ← i
6.  if r ≤ heap-size[A] and a[r] > A[largest]
7.      then largest ← r
8.  if largest ≠ i
9.      then exchange A[i] ↔ A[largest]
10.         MAX-HEAPIFY (A, largest)
```

```
BUILD-MAX-HEAP(A)
1.  heap-size[A] ← length[A]
2.  for i ← ⌊length[A]/2⌋ downto 1
3.      do MAX-HEAPIFY(A,i)
```

5. (a) (5%) Evaluate the postfix expression "40 20 + 30 – 10 *", and show the stack for evaluating this expression before and after processing the '-' operator.
(b) (10%) Write a function *bool isBinaryTree(int preorder[], int inorder[], int n)* to determine if there exists a binary tree of n nodes having the preorder sequence *preorder* [0], *preorder* [1], ..., *preorder*[n-1] and the inorder sequence *inorder*[0], *inorder*[1], ..., *inorder*[n-1].

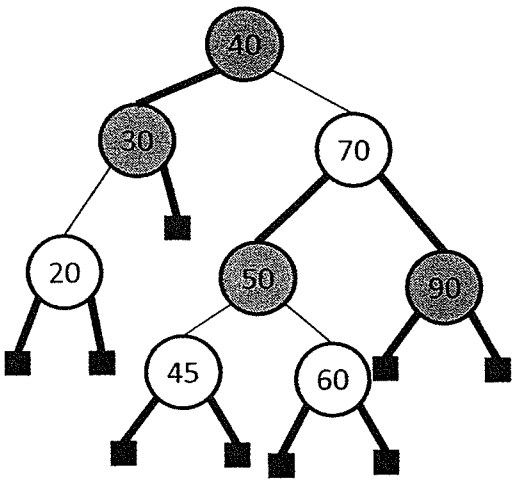
6. (a) (5%) What is the main purpose of threading a binary tree?

(b) (10%) In a right-threaded binary tree, the right link of a leaf node except the right-most leaf node always points to the in-order successor of the node. Write a function *void rightThreading(struct node*root)* for right-threading a binary tree. The node structure of a right-threaded binary tree is defined as follows:

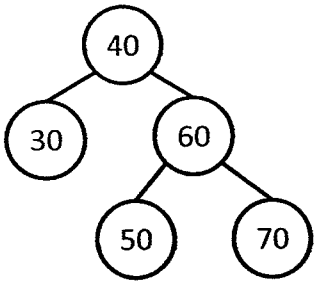
```
struct node {
int data;
struct node *leftChild, *rightChild;
bool rightThread;
};
```

If *rightThread* of a node is ture, *rightChild* of the node is a threaded link; otherwise, *rightChild* of the node is a normal link. Initially, *rightThread* of every node is false.

7. (a) (5%) Depict the following red-black tree after insertion of 65. Notice that the rebalancing rotation and color change are needed to keep the property of the red-black tree.



(b) (5%) Depict the following AVL tree after insertion of 65. Notice that the rebalancing rotation may be needed.



8. (a) (5%) The following is the array representation of a max-heap *H* of seven elements. Show that array after removing the largest element from the max-heap *H*.

H[1]	H[2]	H[3]	H[4]	H[5]	H[6]	H[7]
7	6	5	2	4	1	3

(b) (5%) The following is the array representation of the union-and-find data structure for sets of eight elements 0, 1, 2, ..., 7. Show that array after using the weighting rule to union the sets including elements 2 and 3 together.

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
-3	0	1	-2	3	-1	-1	-1