國立臺灣海洋大學 106學年度研究所碩士班招生考試試題

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. The structure for the node of a linked list is defined as follows:

```
struct list_node {
    int data;
    struct list_node *link;
};
```

where *link* is a pointer to the next node in the linked list.

(a) (10 %) Write a function *int numberOfNodes(struct list_node *head)* that returns the number of nodes in the linked list pointed to by pointer *head*.

(b) (5 %) Fill in the five blanks to complete the function *struct list_node *merge(struct list_node* h1, struct list_node* h2)* that merges two sorted linked lists into one sorted linked list and returns the pointer to the first node of the merged list, where *h1* and *h2* are pointers to the first nodes of the two sorted linked lists.

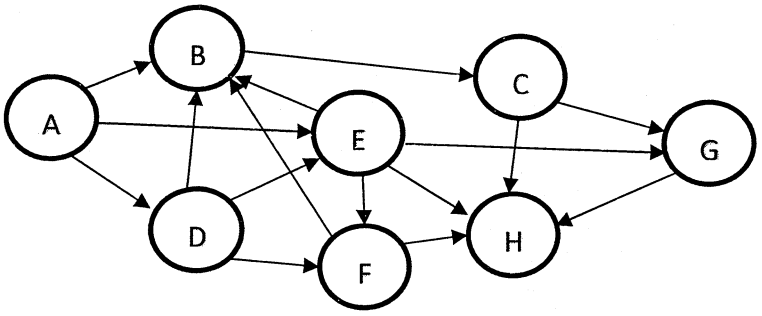
```
struct list_node *merge(struct list_node* h1, struct list_node* h2)
{
    struct list_node*merged, *tail;
    if (h1 == NULL) return h2;
    if(h2 == NULL) return h1;
    if(h1->data < h2->data) {
        merged=h1; h1 = h1->link;
    } else {
        merged=h2; h2 = h2->link;
    }
    tail = merged;
   while(h1 && h2) {
      if(h1->data < h2->data) {
          tail->link =  __(A)__; h1 =  __(B)__;
      } else {
          tail->link =  __(C)__; h2 = __(D)__;
      }
      tail =  __(E)__;
```

```
    }
    if (h1) tail->link = h1;
    if(h2) tail->link = h2;
   return merged;
  }
```

2. (a) (7 %) Convert the following array into a max-heap.

Index	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Value	2	3	0	1	5	6	4

(b) (8 %) Give a topological sort of the following directed graph.

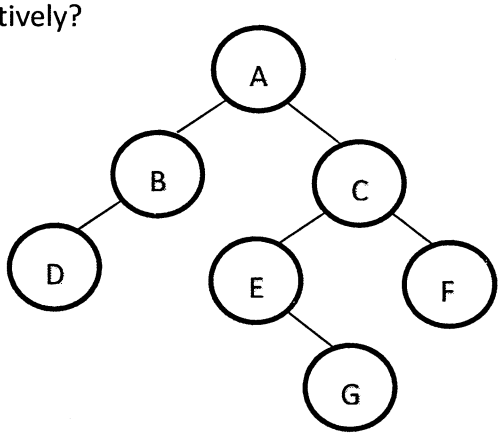


3. The following shows a binary search tree, which contains 7 nodes with distinct keys. The successor of node *x* in a binary search tree is the node with the smallest key greater than the key of node *x*. Note that A, B, ..., G are not keys.

(a) (2 %) Which node is the node with the smallest key?

(b) (4 %) Which node is the successor of node A? Which node is the successor of node B?

(c) (4 %) Suppose that the keys examined for searching for the number 50 in this binary search tree are 30, 80, 40, and 50 in sequence. Which nodes are the nodes with keys 30, 80, 40, and 50, respectively?



(d) (10 %) The structure for the node of a binary search tree is defined as follows:

```
struct tree_node {
    int data;
    struct tree_node *left, *right, *parent;
};
```

where *left*, *right*, and *parent* are pointers to the left child, right child, and parent of a node, respectively. Complete the function

struct tree_node successor(struct tree_node *x),*

which returns the successor of a node *x* in a binary search tree if it exists, and NULL if x has no successor in this tree, by filling in the five blanks.

```
struct tree_node* successor(struct tree_node *x)
{
    struct tree_node *succ;
  if (x == NULL) return NULL;
  if (x->right != NULL) {
      succ = x->right;
      while (__(A)__ != NULL) {
          succ = __(B)__;
      }
  } else {
      succ = x->parent;
      while(succ != NULL && __(C)__ == x) {
          x =__(D)__; succ = __(E)__;
      }
  }
  return succ;
}
```

4. (10%) For a given function $g(n)$, we denote by $\Omega(g(n))$ the set of functions

$O(g(n)) = \{f(n)$: there exist positive constants c and n_0 such that $0 \le cg(n) \le f(n)$ for all $n \ge n_0\}$.

Show that $3n^2 - 4n = \Omega(n^2)$ by demonstrating the constants.

5. (10%) Search trees are data structures that support many dynamic-set operations. For the set of keys {5, 22, 13, 15, 27, 33, 11}, draw binary search trees of height 2, 3, 4, 5 and 6.

6. (15%) An array of n elements contains all but one of the integers from 1 to $n + 1$.

   a. Give the best algorithm you can for determining which number is missing if the array is sorted, and analyze its asymptotic worst-case running time.

   b. Give the best algorithm you can for determining which number is missing if the array is not sorted, and analyze its asymptotic worst-case running time.

7. (15%) Bubble sort is a popular sorting algorithm. It works by repeatedly swapping adjacent elements that are out of order.

```
Bubble-Sort (A)
1. for i = 1 to length[A] – 1
2.       for j = length[A] downto i + 1
3.           if A[j] < A[j – 1]
4.               exchange A[j] ↔ A[j – 1]
```

   a. Illustrate the operation of Bubble-Sort on the array $A = \langle 2, 23, 17, 14, 34, 12 \rangle$.

   b. What is the worst-case running time of Bubble-Sort? How does it compare to the running time of Merge-Sort?