國立臺灣師範大學 110 學年度碩士班招生考試試題

科目：軟體基礎　　適用系所：資訊工程學系

注意：1.本試題共 6 頁，請依序在答案卷上作答，並標明題號，不必抄題。2.答案必須寫在指定作答區內，否則依規定扣分

1. (5 points) Run Time Analysis: Describe the worst case running time of the following pseudocode functions in Big-Oh notation in terms of the variable *n*.
You MUST choose your answer from the following (not given in any particular order), each of which could be the answer for more than one of I. – IV.:

$O(n^2)$, O(n logn), O(n), $O(n^2 \log n)$, $O(n^5)$, $O(2^n)$, $O(n^3)$, O(log n), O(1), $O(n^4)$, $O(n^n)$

a)
```
void func1 (int n, int sum) {
    for (int i = 0; i < n * 100; ++i) {
        for (int j = n; j > 0; j--) sum++;
            for (int k = 0; k < i; ++k)
                sum++;
        }
    }
```

b)
```
int func2 (int n, int m, int sum) {
    if (n < 10)
      return n;
    else {
        for (int i = 0; i < n; ++i)
            sum++;
        return sunny (n - 2, m, sum);
        }
    }
```

c)
```
void func4 (int n, int sum) {
    int j = n;
    while (j > 2) {
        sum++;
        j = j / 2;
        }
}
```

2. (5 points)Give the tightest possible upper bound for the worst case running time for each operation listed below in terms of n. Assume no duplicate values and that you can implement the operation as a member function of the class – with access to the underlying data structure, including knowing the number of values currently stored (n). For array-based implementations, assume that the array is large enough – does not need to be grown.

a) Find minimum value in a Priority Queue implemented with a binary search tree.

b) Pop from a Stack implemented with a linked list.

c) Delete min from a Priority Queue implemented with a binary min heap.

d) Find the maximum value in a binary min heap.

3. Multiple choice questions. (15 points, 3 points for each question)

(3-1) What is the expected number of operations needed to visit all the edges terminating at a particular vertex given an adjacency matrix representation of the graph? (Assume n vertices are in the graph and m edges terminate at the desired node.)

(A) O(m) (B) O(n) (C) $O(m^2)$ (D) $O(n^2)$ (E) $O(m \times n)$

(3-2) What is the expected number of operations needed to visit all the edges terminating at a particular vertex given an adjacency list representation of the graph? (Assume n vertices are in the graph and m edges terminate at the desired node.)

(A) O(m) (B) O(n) (C) $O(m^2)$ (D) $O(n^2)$ (E) $O(m \times n)$

(3-3) Suppose we have an array of ten integers: 5 3 8 9 1 7 0 2 6 4.
Now we use quicksort's partition function and using 5 for the pivot. Which shows the array after partition finishes?

(A) 5 3 4 2 1 0 7 9 6 8

(B) 0 3 4 2 1 5 7 9 6 8

(C) 3 1 0 2 4 5 8 9 6 7

(D) 3 1 0 2 4 5 8 9 7 6

(E) None of the above

(3-4) Which of the following sorting algorithms does not require $O(n^2)$ steps in the worst case?

(A) insertion sort
(B) selection sort
(C) heap sort
(D) bubble sort
(E) quicksort

(3-5) If a binary search tree is not allowed to have duplicates, there is more than one way to delete a node in the tree when that node has two children. One way involves choosing a replacement node from the left subtree. If this is done, which node are we looking for?

(A) the largest node in the subtree
(B) the smallest node in the subtree
(C) the root of the left subtree
(D) the next-to-smallest node in the subtree
(E) it doesn't matter – any node in the left subtree will do

Short Answer questions.

4. (5 points) Consider a hash table with linear probing and a size of 9 buckets with a slot in each bucket. Use the hash function "k%9". Insert the keys: 5, 29, 20, 0, 27 and 18 into your table (in that order). What is the result?

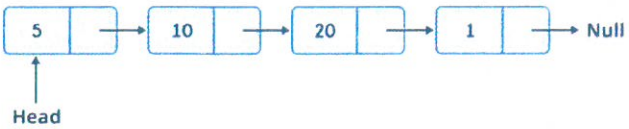
5. (5 points) The array 10 8 6 2 1 4 5 is organized into a heap (priority queue). Please show the array representing the heap after two deleteMax operations have been performed?

6. (5 points) Consider the digraph on eight nodes, labeled 0 through 7, with the 13 directed edges:

0->7 0->1 1->4 1->6 2->3 3->4 4->2 5->2 6->0 6->3 6->5 7->1 7->3.

Please list the strongly connected components.

7. (5 points) Please show the linked list after performing the following code. Also show which nodes are pointed by Head, middle, and trail.




```
listPointer middle, trail;
middle = NULL;
while (Head) {
 trail = middle;
 middle = Head;
 Head = Head->link;
 middle->link = trail;
}
```

8. (5 points) Given the type definition of a node in a threaded binary tree as below.

```
typedef struct thresholdTree *threadPointer;
typedef struct thresholdTree {
  boolean    leftThread, rightThread;
  threadPointer leftChild, rightChild;
  char data;
  };
```

Please fill in the blanks in the codes to find the inorder successor of a node tree in a threaded binary tree without needing a stack?

```
threaded_pointer insucc(threadedPointer tree){
    threaded_pointer temp;
    temp = _____(a)_____;
    if (!tree->rightThread)
     while (_____(b)_____)
         temp = _____(c)_____;
    return temp;
}
```

9. (20 points) Merge-sort and quick-sort solve the sorting problem by using divide-and-conquer. Suppose the input array has n distinct numbers, please answer the following questions regarding the time complexity of each step in these algorithms.

(a) (5 points) Please briefly describe the "divide" step of merge-sort and the time complexity of this step.

(b) (5 points) Please briefly describe the "conquer" step of merge-sort and the time complexity of this step.

(c) (5 points) Please briefly describe the "divide" step of quick-sort and the time complexity of this step.

(d) (5 points) Please briefly describe the "conquer" step of quick-sort and the time complexity of this step.

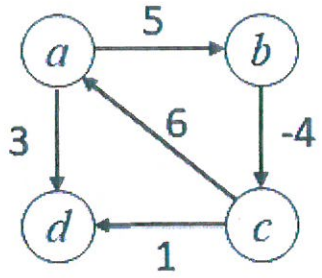
10.(10 points) The Dijkstra's algorithm is used to find the shortest path from a single source vertex to all other vertices in a graph, with the pseudo code given as follows.

```
DIJKSTRA(G, w, s)
// G: input graph, w: edge weights, s: source vertex
1    INITIALLIZE(G, s)
2    S = {}
3    Q = G.V
4    while Q ≠ {}
5        u = EXTRACT-MIN(Q)
6        S = S ∪ {u}
7        for each vertex v ∈ G.Adj[u]
8            if v.d > u.d + w(u, v)
9                v.d = u.d + w(u, v)
```

(a) (5 points) In line 5, this algorithm finds a vertex u hat has a minimal value $u.d$. What does that value mean?

(b) (5 points) Prim's algorithm is designed for solving the minimal spanning tree problem. It is also a greedy algorithm that builds a spanning tree one vertex at a time. Please describe the exact difference between Dijstra's and Prim's algorithms.

11.(20 points) Given the graph below, we apply Johnson's algorithm (a re-weighting method) to find the shortest paths between all pairs of vertices. Please answer the following questions regarding each step of Johnson's algorithm.



(a) We first add a new node q to the input graph, and connect it with zero-weight edges to each of the other nodes. Next, we run the Bellman-Ford algorithm to find for each vertex $v \in \{a, b, c, d\}$ the minimum weight $h(v)$ of a path from q to v. The Bellman-Ford algorithm makes several passes over all edges of the graph. Please explain:

i. (3 points) Why the Bellman-Ford algorithm is used, rather than using Dijkstra's algorithm?

ii. (2 points) How many passes do we need for processing the graph above (including the augmented vertex q)?

(b) (5 points) The previous step returns the results: $h(a) = 0$, $h(b) = 0$, $h(c) = -4$, and $h(d) = -3$. An edge from u to v, having a length $w(u, v)$, is now given the new length $w(u, v) + h(u) - h(v)$. This step clearly reweights the edges of the original graph. Please draw the reweighted graph.

(c) (5 points) Finally, we apply Dijkstra's algorithm to find the shortest paths from each node to every other vertex in the reweighted graph. Please write down the shortest path lengths using vertex a as the source. In other words, please compute the shortest path length $\delta(a, b)$ between a and b, and similarly, $\delta(a, c)$ and $\delta(a, d)$.

(d) (5 points) Professor Smith says that there is a simpler way than Johnson's method to reweight edges. Let $c = \min\{w(u, v)\}$, and define $\hat{w}(u, v) = w(u, v) - c$ for all edges (u, v) of the input graph. What is wrong with Smith's reweighting method for solving the all-pairs shortest paths problem?