國立交通大學 106 學年度碩士班考試入學試題

科目：資料結構與演算法(1101)　　考試日期:106年2月10日 第 1 節

系所班別：資訊聯招　　第 1 頁,共 3 頁

【不可使用計算機】＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

1. [5%] Let $p = p_0 p_1 \cdots p_{n-1}$ be a pattern. Consider the following definition of failure function:

$$f(i) = \begin{cases} \text{The largest } i < j \text{ such that } p_0 p_1 \cdots p_i = p_{j-i} p_{j-i+1} \cdots p_j \text{ and } p_{i+1} \neq p_{j+1} \\ -1, \text{ if there is no } i \geq 0 \text{ satisfying above} \end{cases}$$

Show the values of $f(4)$, $f(5)$, $f(6)$, $f(7)$ and $f(8)$ for the pattern **abcabcacab**.

2. [5%] Please answer the following questions.

(a) [2%] Show the prefix expression of the infix expression: **A + B * (C – D) / E**

(b) [2%] Show the postfix expression of the infix expression: **A + B * (C – D) / E**

(c) [1%] Show the result of the prefix expression: *** 3 – 6 + 2 2**

3. [5%] Please show the result of the following program:

```
#include <iostream>
using namespace std;
struct node
{
  int value;
  struct node * next;
};
struct node n1, n2, n3, n4;
void f(struct node *p) {
  struct node * q;
  int t;
  if (p == NULL) return;
  for (q = p;  p->next != NULL;  p = p->next)
    for (q = p;  q->next != NULL;  q = q->next)
      if ( (q->value % 2 == 0) && (q->next->value < q->value) ) {
        t = q->next->value;
        q->next->value = q->value;
        q->value=t;
      }
}
int main(void)
{
  struct node * p;
  n1.value=8;  n2.value=3;
  n3.value=2;  n4.value=7;
  n1.next=&n2; n2.next=&n3;
  n3.next=&n4; n4.next=NULL;
  f(&n1);
  for(p=&n1; p!=NULL; p=p->next)
    cout << p->value << " ";
  return 0;
}
```

4. [5%] Please answer the following questions.

(a) [2%] Please give the definition of min heap.

(b) [3%] Consider an empty min heap. We first insert the numbers, **18, 10, 4, 8, 2, 15, 6, 9**, into the min heap. We then perform deletion twice. Please draw the resultant min heap.

5. [5%] Consider a binary tree consisting of six nodes: **A**, **B**, **C**, **D**, **E** and **F**. Given the following inorder and postorder traversal sequences of the tree, please draw the tree.

Inorder traversal sequence: **ABDCFE**

Postorder traversal sequence: **ADFECB**

6. [5%] Consider a 13-slot hash table for storing English words. Each slot can hold one word. The hash address of a word is determined by its first letter, with letters **A** to **M** indexed to slots 0-12, respectively, and letters **N** to **Z** also indexed to slots 0-12, respectively. Now, starting from an empty hash table, insert the following words in the given order:

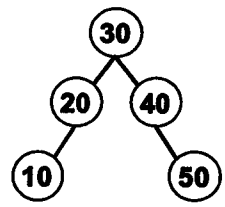
JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC

(a) [3%] If collisions are handled using open addressing with linear probing, indicate the indices of the slots that hold the following three words: **MAR, AUG, DEC**

(b) [1%] If collisions are handled using chaining, what is the maximum chain length in any slot?

(c) [1%] Following (b), list all the slots that have the maximum chain length.

7. [10%] Answer the following questions about balanced binary search trees:



(a) [2%] Draw the resultant AVL tree after inserting a new item with key=**15** to the AVL tree shown to the right.

(b) [2%] Draw the resultant AVL tree after inserting a new item with key=**60** to the AVL tree shown to the right.

(c) [6%] Insert the following keys into an initially empty red-black tree in the given order:

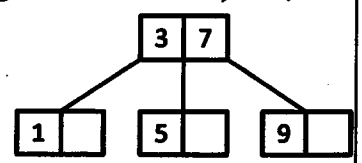
1, 3, 5, 2, 4, 6.

Draw the resultant red-black tree. When drawing, use circles for red nodes and squares for black nodes. You do not need to draw the external nodes.

8. [6%] Answer the following questions about B-trees:

(a) [1%] According to the definition of B-trees, if the order (the maximum branching factor of a node) is 6, what is the maximum number of items that can be held in a B-tree with three levels? Just give your answer; no computation required.

(b) [5%] Starting with the order-3 B-tree to the right, insert these keys: **2, 4, 6**, to the tree in the given order. Draw the resulting B-tree.



9. [4%] Consider a Fibonacci heap used as a Min priority queue. For each of the four operations below, answer yes or no regarding whether it has $\Theta(1)$ time complexity.

(a) **Insert**　(b) **Extract-Min**　(c) **Decrease-Key**　(d) **Union (the merge of two priority queues)**

10. [10%] Prove that the average-case time complexity for Quicksort is $O(n \log n)$.

11. [10%] Design a Greedy algorithm to generate an optimal 2-way merge tree.

Input: m sorted lists, L_i, i=1, 2, ⋯, m, with each L_i consisting of n_i elements.

Output: An optimal 2-way merge tree.

12. [3%] For each statement below, indicate whether it is correct or not.
(S1) If any NP-complete problem can be solved in polynomial time, then NP = P.
(S2) If problem A is an NP problem, there must exist an NP algorithm B which solves A.
(S3) Since the satisfiability problem is NP-complete, the 2-satisfiability problem is also NP-complete.

13. [2%] Let

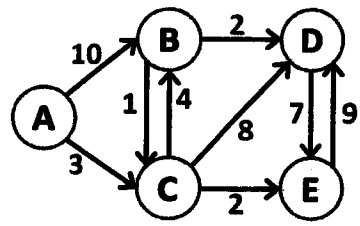
$$T(n)=\begin{cases} b, & \text{for } n=1, \\ aT(n/c)+bn, & \text{for } n>1, \end{cases}$$

where a, b, and c are nonnegative constants and n is a power of c. Give $T(n)$ as a big-theta expression for a=3 and c=2.

14. [10%] Let G = (V, E) be a connected, undirected graph with edge-weight function w:E→R, and assume that all the edge weights are distinct. Consider a cycle $(v_1, v_2,\ldots,v_k, v_{k+1})$ 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.

15. [5%] A graph G has edges which are colored either red or blue. Give the fastest algorithm that you can think of to compute a spanning tree with as few blue edges as possible.

16. [5%] Execute Dijkstra's algorithm on the graph shown to the right, starting at vertex **A**. If there are any ties, the vertex with the lower letter comes first. List the vertices in the order in which they are deleted from the priority queue. For each vertex, also show the shortest distance from **A** to that vertex.



17. [5%] The residual graph shown below is for some flow f on a flow graph G.
Find a shortest augmenting path. Draw the residual graph that results from adding as much flow as possible to this path.