Mid-Term Exam of ***Data Structures*** (Fall 2018)
Department of Information Management and Finance
National Chiao Tung University

The answers should be brief and precise in C/Java-like pseudo codes. The input and output of any algorithm should be specified explicitly.

PART 1: True or False (Explain the reason if your answer is FALSE.) (10 pts.)

1. (T) Notation big-O aims at measuring the upper bound of a function, while Ω focuses on the lower bound.
2. (T) In C language, the memory locations allocated for a static 2-D memory are consecutive; whereas, those for a dynamic allocation are not.
3. (F) N. Wirth is well known by his book *Algorithms + Data Structures = Programs*, and received the Turing Award for his contribution in computer security.
4. (F) If there is an algorithm A solving P, which has a lower bound $\Omega(g(n))$, in time $O(g(n))$ in the worst case, then A is an optimal algorithm for P. Thus, the optimal algorithm of P is unique (which is exactly A).
5. (F) The “undo” (Ctrl-Z) capability of many software applications could be implemented by using a queue.
6. (T) $4096 = \Theta(1)$
7. (F) $12n^3+\log n = O(n^2)$
8. (F) $n^3\log n = \Theta(n^3)$
9. (F) $\sum_{i=0}^{n} i^2 = O(n^2)$
10. (T) In C, &data[i][j] is equivalent to data[i]+j and *(data+i)+j.

PART 2: Answer the following questions:

1. Suppose that n integers are stored in 1D array A. We would like to search target x in A and expect index k if $A[k] == x$, or -1 otherwise (x cannot be found in A).
   (a) (2 pts.) Design a *linear search* algorithm to search x in A.
   (b) (4 pts.) Design a *binary search* algorithm to search x in B where B is another array consisting of all data in A in the non-decreasing order.
   (c) (1 pts.) What are the time complexities of (a) and (b)?

(d) (3 pts.) Let the number of searches be k. Based on (c), give your suggestion on when to applying linear and binary search algorithms.

2. Determine the time complexity of the following pieces of programs using the tightest big-O notation.

(a) (3%)
```
for (w=1; w<=n; w++)
    for (x=n; x>=1; x/=2)
        for (y=1; y<=x; y++)
            z++;
```

(c) (3%)
```
for (x=1; x<=n; x*=2)
    for (y=x; y>=1; y/=2)
        z++;
```

3. A polynomial of degree k in a single indeterminate x is denoted as

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_kx^{k-1}.$$

Suppose that the coefficient $a_0, a_1, a_2, \dots, a_k$ have been stored in 1-D array data; that is, data[i] = a_i for $0 \le i \le k$.

(a) (6 pts.) Design an algorithm to compute $f(1), f(2), \dots, f(n)$.

(b) (2 pts.) How many multiplications ($\times$) and additions (+) are there in (a)?

4. (a) (3 pts.) Design a recursive algorithm to compute the Ackermann function:

$$A(m,n) = \begin{cases} n-1 & \text{if } m=0; \\ A(m-1,1) & \text{if } n=0; \\ A(m-1, A(m, n-1)) & \text{otherwise.} \end{cases}$$

(b) (4 pts.) Fill the blank entries of the table, which are the values of $A(m, n)$'s.

$m \setminus n$	0	1	2
0			
1			
2			

5. (a) (2 pts.) Given the magic square where "1" has been predefined at the center of the left-most column.

1				

(b) (2 pts) What is the summation value of each row/column/diagonal of an 11×11 magic square?

6. (5 pts.) Prove that the minimum steps of moving disks in the "Towers of Hanoi" problem is 2^n-1.

7. Consider two row-major arrays A and B. Let the starting address be α and β, respectively, and the size of each element in A or B be l.
   (a) (2 pts) Find the address of $A[i][j]$ in terms of α and l, where A is a $u_1 \times u_2$ 2-D array.
   (b) (3 pts) Find the address of $B[i_1][i_2]$ ... $[i_n]$ in terms of β and l, where B is a $u_1 \times u_2 \times \ldots \times u_n$ n dimensional array.
   (c) (3 pts) How many multiplications and additions are needed to compute the address in (b).

8. (3 pts.) Let SWAP(a, b, c) be a macro performing the swapping of values between parameters a and b. Function p is defined as follows.

```
void p(char *list, int k, int n)
{   int i, tmp;
    if (k == n-1) cout << list << endl; //output list
    else
    { for (i=k; i<n; i++)
       { SWAP(list[k], list[i], tmp); // list[0]: the first element
          cout << "=>" << i << ", " << k << ", " << list << endl;
          p(list, k+1, n);
          SWAP(list[k], list[i], tmp);
          cout << "<=" << i << ", " << k << ", " << list << endl;
       }
    }
}
```

Let list = "ABCD"; be set in the main program. What would be printed out after function p(list, 2, 4); is called?

9. Let A^B denote A^B in which the priority of ^ is higher than that of * and /.
   (a) (A+B)*C^(D+E*F-G*(H+J))
   (2 pts.) Give the postfix notation of (a). (AB+CDEF*+GHJ+*-^*)
   (b) QRSTUVWX*-^*Y/+*+
   (2 pts.) Give the prefix notation of (b). (+Q*R+S/*T^U-V*WXY)

Suppose that functions “push” and “pop” for strings have already been defined. You could call them directly. Please design algorithms for

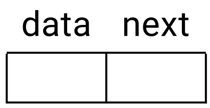
(c) (5 pts.) transforming the infix notation into the postfix notation;

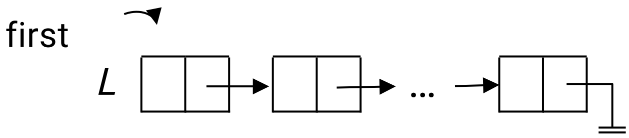
(d) (3 pts.) transforming the postfix notation into the prefix notation;

(d) (3 pts.) transforming the prefix notation into the postfix notation;

10. Based on the following declaration, a singly linked list *L* would be constructed as shown below.

```
struct node
{    int data;
     struct node * next;
};
struct node    * first;
```





(a) (3 pts.) Design the following function to insert a new node storing an input integer named element before the node pointed by first. (Hint: *x* may be a null pointer NULL.)

```
struct node * insertFront(struct node * x, int element)
```

(b) (3 pts.) Design a function to insert a new node storing an input integer named element at the end of the list.

```
struct node * insertRear(struct node * x, int element)
```

(d) (3 pts.) Given a pointer *x* pointing to a certain node in *L*, design a function to delete the node, whose data should be returned, after the one pointed by *x*. (Hint: *x* may be first, or a null pointer NULL)

```
int deleteAfter(struct node * x)
```

11. (10 pts.) Consider the following program segments.

[A]
```
int data[10][10], i, j;
for (i = 0; i < 10; i++)
    for (j = 0; j < 10; j++)
        data[i][j] = 1000-10*i-j-1;
```

Suppose that we obtain 6606000 when (int) &data[0][0] is printed and we have 999 when data[0][0] is printed. What values would be printed when the following variables are printed as integers?

(a) data[1]

(b) (int)data[1]+1

(c) data[1][0]

(d) &data[1][1]

(e) *(data[0]+1)

[B]
```
int ** data, i, j;
data = new int * [10];
for (i=0; i<10; i++) data[i] = new int [n];
for (int i=0; i<m; i++)
        for (j = 0; j < n; j++)
                data[i][j] = 100+10*i+j;
```

Suppose that we obtain 3318000 when (int) &data[0][0] is printed,
3318048 (int) &data[0][0]
3318096 (int) &data[1][0]
100 data[0][0]

What the values would be when the following variables are printed as integers?

(f) data[1]+1
(g) data[1][0]
(h) *(data[0]+1)
(i) &data[1][1]
(j) *(*(data+1)+1)

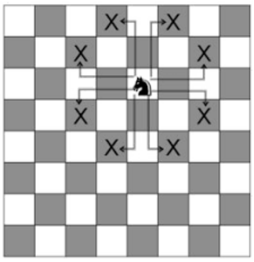
12. A palindrome is a word or group of words that is the same when you read it forwards from the beginning or backwards from the end. For instance, "top spot", "Eva, can I see bees in a cave?" (by ignoring the punctuation and spaces), "上海水自來水來自海上", "遙望四邊雲接水 碧峰千點數鷗輕 輕鷗數點千峰碧 水接雲邊四望遙" are palindromes. Design an algorithm to determine whether input string *S* is a palindrome by
(a) (4 pts.) using a stack;
(b) (2 pts.) using a simple loop;
(c) (3 pts.) using recursion.

13. Consider an $m \times n \times p$ 3 dimensional 0/1 maze with one entrance at (1, 1, 1) and one exit at (m, n, p) where 0's entries represent open paths while 1's the barriers. The mouse-maze problem aims at simulating the situation for a mouse to find a tour from the entrance to the exit by a computer program.
(a) (4 pts.) Specify your data structures and explain their purposes.
(b) (5 pts.) Design an algorithm to find a tour (or report no such a tour) using the trail and error strategy.
(c) (2 pts.) Refine your algorithm in (b) so that all tours in the maze can be explored. (Hint: Simply state precisely your refinement.)

(d) (2 pts.) What are the time complexities of your algorithms in (a) and (b), respectively.

14. In a classic chessboard, the possible moves of a knight is as Fig. 1(a). Given an $n \times n$ board B, a *knight's tour*, starting from any square, traverses all squares on B by knight's moves. Let each square be indexed as j when the knight takes the jth move here. Specifically, a tour 1-2-3- ... -n^2 on B is a knight's tour if i-(i+1) is a legal knight's move for $1 \le i \le n^2 - 1$. Fig.1(b) shows 3 knight's tours in a 5×5 board and Fig. 1(c) gives a knight's tours on a 8×8 board.

(a)



(b)

1	22	13	18	7
14	19	8	23	12
9	2	21	6	17
20	15	4	11	24
3	10	25	16	5

1	14	9	20	3
24	19	2	15	10
13	8	25	4	21
18	23	6	11	16
7	12	17	22	5

1	14	9	20	3
24	19	2	15	10
13	8	23	4	21
18	25	6	11	16
7	12	17	22	5

(c)

0	59	38	33	30	17	8	63
37	34	31	60	9	62	29	16
58	1	36	39	32	27	18	7
35	48	41	26	61	10	15	28
42	57	2	49	40	23	6	19
47	50	45	54	25	20	11	14
56	43	52	3	22	13	24	5
51	46	55	44	53	4	21	12

Fig.1 Knight's moves and knight's tours

(a) (4 pts.) Design an algorithm to determine whether the n^2 indices on the n^2 squares of a given $n \times n$ board B constitute a knight's tour.

(b) (1 pts.) What is the time complexity of your algorithm?

15. (4 pts.) Fig. 2 shows a switching network (stack) for train carts with stack size 2 (where only two carts, say *A* and *B* as shown, could be pushed into the stack). Each cart in the left side should be pushed into the stack once and popped out once to the right side. List all possible permutations (in the right) of the four carts, which are in the left initially and named as *A*, *B*, *C* and *D* from right to left.

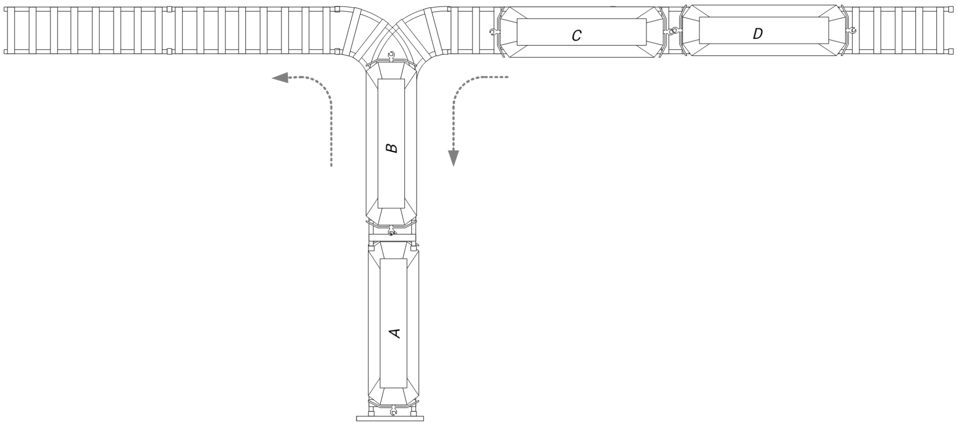


Fig.2 Switching network for train carts with stack size 2

- END -