Data Structures
Midterm Exam
Sketch Solution

Name: ______________________________

ID #: ______________________________

This is a Close Book examination. You can write your answers in English or Chinese.

	Maximum	Score
Problem 1	15	15
Problem 2	12	12
Problem 3	8	8
Problem 4	15	15
Problem 5	5	5
Problem 6	10	10
Problem 7	15	15
Problem 8	10	10
Problem 9	10	10
Total	100	100

1. (15 pts) Mark by T(=True) or F(=False) each of the following statements. You don't need to prove it.
判斷下列敘述，標示T表示為真，標示F表示為假。本題無需證明。

   (1) (3 pts) If $d(n)$ is $O(f(n))$ and $f(n)$ is $O(g(n))$, then $d(n)$ is $O(g(n))$.
   (2) (3 pts) $2^n = O(n!)$.
   (3) (3 pts) n^2 is $\Omega(n \log n)$.
   (4) (3 pts) $\sum_{i=1}^{n} i^2 = O(n^3)$.
   (5) (3 pts) $n^n = O(2^n)$.

Question	Answer
(1)	T
(2)	T
(3)	T
(4)	T
(5)	F

2. (12 pts) Please find the order of growth of the followings:
請找出下列式子的成長速率。

   (1) (3 pts) $n^3 2^n + 6n^2 3^n$
   (2) (3 pts) $\sum_{i=1}^{n} i^3$
   (3) (3 pts) $\sum_{i=2}^{n-1} \log i^2$
   (4) (3 pts) $\sum_{i=1}^{n} (i+1) \times 2^{i-1}$

   Solution:

Question	Answer
(1)	$O(n^2 3^n)$
(2)	$O(n^4)$
(3)	$O(n \log n)$
(4)	$O(n 2^n)$

3. (8 pts) For the following pairs of functions, find the smallest integer value of $n > 1$ for which the first becomes larger than the second.
對下列兩對函式，分別找出最小整數$n > 1$使得第一個函式值會大於第二個函式值。

   (a) (4 pts) 2^n and $8n^4$
   (b) (4 pts) $0.1n$ and $10 \log n$

Solution:

   (a) $n = 21$
   (b) $n = 997$

4. (15 pts) Consider the following algorithm **Enigma** and answer the related questions.
參考以下的演算法Enigma並回答相關問題。

```
def Enigma(A):
    \# Input: An nxn two dimensional matrix A of real numbers
    for i in range(len(A)):
        for j in range(i+1, len(A[i])):
            if (A[i][j] != A[j][i]):
                return False
    return True
```

(1) (3 pts) What does algorithm `Enigma` compute?
演算法Enigma所運算的目的為何?

(2) (3 pts) What is the main (basic) operation (costing the most time)?
哪種運算是最費時的?(即主要操作)

(3) (3 pts) How many times is the main operation executed?
主要操作(或其所屬的敘述)總共做了幾次?

(4) (3 pts) What is the time complexity of algorithm `Enigma`?
演算法Enigma的時間複雜度為多少?

(5) (3 pts) Suggest an improvement or better algorithm altogether and indicate its time complexity. If you can not do it and think it is not possible, please show or explain this is the best and it can not be improved.
請提供一個更快的作法並指出其時間複雜度，若找不到一個更快的作法並且覺得不可能找到，請說明理由並解釋為何這個方法應該是最好的。

solution

(1) Algorithm `Enigma` returns ”true” when the input matrix is symmetric and ”false” if it is not.

(2) Comparison of two matrix elements

(3) The number of times the main operation executed, $C(n)$, can be computed as

$$\begin{aligned} C(n) &= \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 \\ &= \sum_{i=0}^{n-2} [(n-1) - (i+1) + 1] \\ &= \sum_{i=0}^{n-2} (n-1-i) \\ &= (n+1) + (n+2) + \cdots + 1 \\ &= \frac{(n-1)n}{2} \end{aligned}$$

(4) According to (3), the time complexity of algorithm `Enigma` is quadratic, $O(n^2)$.

(5) This algorithm is optimal because any algorithm that solves this problem must, in the worst case, compare $\frac{(n-1)n}{2}$ elements in the upper-triangular part of the input matrix with their symmetric counterparts in the lower-triangular part, which is all this algorithm does.

5. (5 pts) Recall the ***Pascal's triangle*** we considered in our homework set 2. The binomial coefficients may be defined by the following recurrence relation.
回顧作業中所介紹的***Pascal's triangle***，二項式係數可以用下面遞迴的方式定義。

$$\begin{array}{ll} C(n,0)=1 \quad \text{and} \quad C(n,n)=1 & \text{for } n \geq 0 \\ C(n,k)=C(n-1,k)+C(n-1,k-1) & \text{for } n>k>0 \end{array}$$

Please show that the recurrence is correct.
請證明此遞迴定義中遞迴式的正確性。

solution

$$\begin{aligned} C(n,k) &= \frac{n!}{(n-k)!k!} \\ &= \frac{n(n-1)!}{(n-k)!k!} \\ &= \frac{(n-k+k)(n-1)!}{(n-k)!k!} \\ &= \frac{(n-k)(n-1)!}{(n-k)!k!}+\frac{k(n-1)!}{(n-k)!k!} \\ &= \frac{(n-1)!}{(n-k-1)!k!}+\frac{(n-1)!}{(n-k)!(n-1)!} \\ &= C(n-1,k)+C(n-1,k-1) \end{aligned}$$

6. (10 pts) Consider the Fibonacci sequence $\{f_n\}$. Use mathematical induction to show that for $n \geq 1$,

$$\sum_{i=1}^{n} f_i^2 = f_n f_{n+1} - 1,\ n \geq 1.$$

Please note that $f_0 = 1,\ f_1 = 1,\ f_2 = 2,\ f_3 = 3,\ f_4 = 5,\ ...$ and the general definition for the Fibonacci sequence is $f_i = f_{i-1} + f_{i-2},\ i \geq 3$.
考量到一個費氏數列 $\{f_n\}$，請用數學歸納法來證明對所有$n \geq 1$，下式成立

$$\sum_{i=1}^{n} f_i^2 = f_n f_{n+1} - 1,\ n \geq 1.$$

注意，$f_0 = 1,\ f_1 = 1,\ f_2 = 2,\ f_3 = 3,\ f_4 = 5,\ ...$而費氏數列的定義為 $f_i = f_{i-1} + f_{i-2}$, $i \geq 3$。

Solution:

Base Case: when $n = 1$, $f_1^2 = 1 = f_1 \times f_2 - 1 = 1 \times 2 - 1 = 1$.
Assumption: we assume that $\sum_{i=1}^{k} f_i^2 = f_k f_{k+1} - 1,$ for $k \geq 2$.
Induction Step:

$$\begin{aligned}
\sum_{i=1}^{k} f_{k+1}^2 &= \sum_{i=1}^{k} f_i^2 + f_{k+1}^2 \\
&= (f_k f_{k+1} - 1) + f_{k+1}^2 \\
&\quad \text{(Note: using the assumption)} \\
&= ((f_{k+2} - f_{k+1}) f_{k+1} - 1) + f_{k+1}^2 \\
&\quad \text{(Note: } f_k = f_{k+2} - f_{k+1}\text{, according to the definition)} \\
&= (f_{k+1} f_{k+2} - f_{k+1}^2 - 1) + f_{k+1}^2 \\
&= f_{k+1} f_{k+2} - 1 \\
&\quad \text{(Note: crossing out } f_{k+1}^2 \text{ terms)}
\end{aligned}$$

Hence,

$$\sum_{i=1}^{k} f_{k+1}^2 = f_{k+1} f_{k+2} - 1.$$

By Mathematical Induction, we have done the proof.

7. (15 pts) Please answer the following problems simply.

(a) (5 pts) Please write the following infix expression in postfix form:
請寫出下面中序表示式的後序表示式。

$$(a - b * c/d * e + f - g) * h$$

(b) (5 pts) Please write the following prefix expression in infix form with parentheses:
請寫出下面前序表示式的中序表示式。

$$/ * 4a - */ * bc + abdc$$

(c) (5 pts) Draw the sequence of stack configurations for the evaluation of the following postfix expression. Assume that $a = 2$, $b = 3$, $c = 8$, $d = 5$, and $e = 6$.
一步一步畫出當計算下面後序運算式時所使用的堆疊的變化情形。

$$ab * ca - /de * +$$

Solution:

(a) $abc * d/e * -f + g - h*$
(b) $4 * a/(b * c/(a + b) * d - c)$
(c) The output is 33.

Operation	**Contents of Stacks**
`push`(2)	2
`push`(3)	2,3
`pop`	2
`pop`	-
`push`(2*3)	6
`push`(8)	6,8
`push`(2)	6,8,2
`pop`	6,4
`pop`	6
`push`(8-2)	6,6
`pop`	6
`pop`	-
`push`(6/6)	1
`push`(5)	1,5
`push`(6)	1,5,6
`pop`	1,5
`pop`	1
`push`(5*6)	1, 30
`pop`	1
`pop`	-
`push`(1+30)	31
`pop`	-

8. (10 pts) Suppose that data items numbers 1, 2, 3, 4, 5, 6 come in the input stream in this order. That is 1 comes first, then 2, and so on. By using (1) a ***queue*** and (2) a ***deuqe***, which of the following rearrangements can be obtained in the output order? (Yes(Y) or No(N))
假設資料項目依照編號1、2、3、4、5、6順序放入，下列輸出順序是否可利用(1)queue或(2)deque產生? Y表示可以，N則否。

(a) 123456 (b) 526341 (c) 152436

Solution:

arrangement	output order	Queue	Deque
(a)(2 pts)	123456	Y	Y
(b)(4 pts)	526341	N	N
(c)(4 pts)	152436	N	Y

9. (10 pts) Describe a fast recursive algorithm for reversing a singly linked list L, so that the ordering of the nodes becomes opposite of what it was before.
請描述一個快速遞迴演算法來反轉一個單向鏈結串列。

solution

Let us define a method $\mathsf{reverse}(L, n)$, which reverses the first $n \leq L.\mathsf{size}()$ nodes in

L, and returns a pointer end to the node just after the nth node in L ($end =$ **null** if $n = L.\mathsf{size}()$). If $L.\mathsf{size}() \leq 1$, we are done, so let us assume L has at least 2 nodes. If $n = 1$, then we return $L.\mathsf{first}().\mathsf{next}()$. Otherwise, we recursively call $\mathsf{reverse}(L, n-1)$, and let end denote the returned pointer to the nth node in L. We then set ret to $end.\mathsf{next}()$ if $n < L.\mathsf{size}()$, and to **null** otherwise. We then insert the node pointed to by end at the front of L and we return ret. The total running time is $O(n)$.