Data Structure and Algorithm, Spring 2018
Homework #1

Release: Tuesday, March 13, 2018
Due: Tuesday, March 29, 2018
TA E-mail: dsa1@csie.ntu.edu.tw

Non-Programming problems

Problem 1. Stack/Queue

You have learned in class that Stack is a FILO/LIFO data structure and that Queue is a FIFO/LILO data structure. Please design an algorithm(pseudo code) under the followings scenarios. They are all independent. You can assume that all data structure mentioned below have infinite capacity and all operations are valid. That is, no `Pop()` will be called on an empty stack and no `DeQueue()` will be called on an empty queue.

a. Please use 2 queues and $O(1)$ extra space to simulate a stack. That is, please implement `Pop()`, `Push()` and `IsEmpty()` of a stack.

b. Please use 2 stacks and $O(1)$ extra space to simulate a queue. That is, please implement `DeQueue()`, `EnQueue()` and `IsEmpty()` of a queue.

You are given a stack S and a queue Q. They both have infinite capacity, and they both store positive integers. The numbers of elements of S and Q will both be not greater than n. Please describe a procedure to determine whether there exists common element between S and Q in the following scenarios. If there exists, please output arbitrarily one of the common elements. If not, please output 0. The scenarios are independent, and you can answer them in any order.

c. Guaranteed that the elements in S, from bottom to top, are strictly increasing, and the elements in Q, from rear to front, are also strictly **increasing**. In $O(n)$ time, $O(1)$ extra space.

d. Guaranteed that the elements in S, from bottom to top, are strictly increasing, while the elements in Q, from rear to front, are strictly **decreasing**. In $O(n)$ time, $O(1)$ extra space.

Problem 2. Complexity

In this part, if you would like to use any theorem which is not mentioned in class, please prove it in advance.

$$\lg(n) = \log_2(n),\ \ln(n) = \log_e(n),\ \log(n) = \log_{10}(n)$$

1. Please rank the following functions by the order of growth. No proof or extra statement is needed.

$$4^n, 8^n, n^2, n^n, n \lg n, e^{\ln(n)}, 2^{\sqrt{\lg n}}$$

2. Prove that $\log(n!) = O(n \lg n)$

3. Prove or disprove the following statements. You should provide a formal proof or a counterexample for each statement.

   a. $100n^2 + 300n = O(n^2)$.

   b. If $f(n) = O(i(n))$, $g(n) = O(j(n))$, then $f(n) - g(n) = O(i(n) - j(n))$.

   c. If $f(n) = O(g(n))$, then $f^2 = O(g^2)$.

   d. If $f(n) = \Omega(i(n))$, $g(n) = \Omega(j(n))$, then $f(n) + g(n) = \Omega(i(n) + j(n))$

   e. If $f(n) = O(g(n))$, then $2^{f(n)} = O(2^{g(n)})$

   f. If $2^{f(n)} = O(2^{g(n)})$, then $f(n) = O(g(n))$

   g. $f(n) + g(n) = \Theta(\max(f(n), g(n)))$

4. Define a function $f(n)$ as below:

$$f(n) = 1,\ \text{when } n = 1$$
$$2f([\frac{n}{2}]) + n,\ \text{otherwise}$$

   Please find out the tightest bound of $f(n)$ with Θ notation. You should provide a proof to get credits. Answers without any proof will NOT get any credits. (Note: If you want to use the Master Theorem, then you should prove it first)

5. Define a function $T(n)$ as below:

$$T(n) = 1,\ \text{when } n = 1$$
$$2T([\sqrt{n}]) + \lg n,\ \text{otherwise}$$

   Prove that $T(n) = \Theta(\lg n \times \lg \lg n)$

Problem 3. Linked List.

1. You are given the heads of two read-only linked list A and B. Both A and B are loop-free and store positive integers. The integers in A, from head to tail, form a sequence $\langle a \rangle$, and the integers in B, from tail to head, form a sequence $\langle b \rangle$. Guaranteed that $\langle a \rangle$ and $\langle b \rangle$ are of same length n, and $\exists! i$ s.t. $a_i \neq b_i$. Please describe a procedure to find i.


$\langle a \rangle = (9, 4, 8, 7), \langle b \rangle = (5, 4, 8, 7), i = 1, a_i = 9, b_i = 5.$
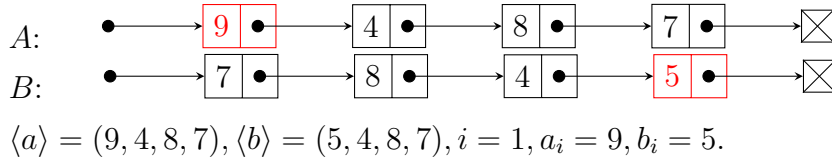
Your procedure should take the heads of A and B as its input, then output i, while i is the index mentioned above. Both A and B are read-only, you cannot modify them.

In $o(n^2)$ time, $O(1)$ extra space.

2. You are given the heads of two read-only linked list A and B. They might contain loop. Please follow the guide and design a procedure to determine whether the shape of the linked lists are identical.

For example, among the following linked lists, linked list β and linked list δ are identical in shape. You may discover that a linked list contains at most one loop, and if there is one, then the loop must appears at the end. Thus, two linked list are identical in shape if and only if their "chain part" and the "loop part" are of equal size respectively.

3

ϵ:
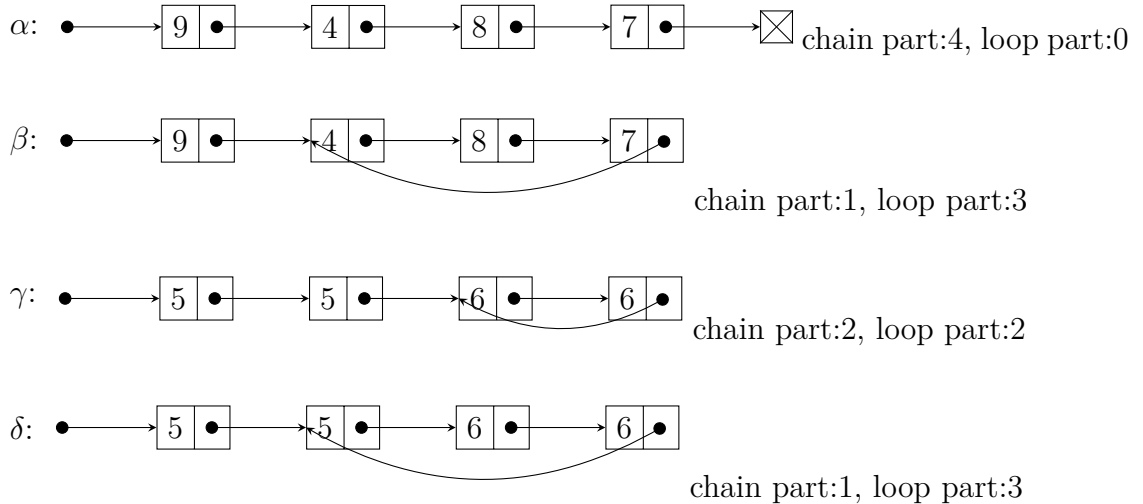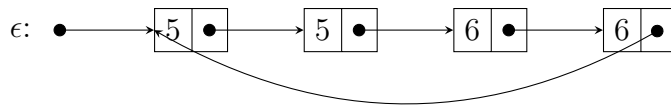
chain part:0, loop part:4

Both A and B are read-only, you cannot modify them.

a. Please complete the C function below to determine whether the given linked list has a loop. If so, return a pointer to an arbitrary node on the loop. Otherwise, return *NULL*.

In $O(n)$ time, $O(1)$ extra space, while n is the number of nodes of the given linked list. Please note that the only input is the head of the linked list, and you are ignorant of n.

```
typedef struct list_node{
    void* data;
    struct list_node *next;
} list_node;
*list_node isLoop(list_node *head){
    //fill the blank here
}
```

b. Please complete the C function below to determine the length of the loop of the given linked list. The length of the loop is defined as the number of nodes on the loop.

In $O(n)$ time, $O(1)$ extra space.

```
unsigned int loopLength(list_node *head){
    //fill the blank here
}
```

Please describe a procedure that determine whether A and B are of same shape. Your procedure should takes the heads of A and B as its input. If they are of same shape, output 1, otherwise, output 0. In:

c. $O(n \lg n)$ time, $O(1)$ extra space.

d. $O(n)$ time, $O(n)$ extra space.

n is the sum of the numbers of nodes of A and B, and you are ignorant of n.

Programming problems

Problem 4. Notepad --

You are required to simulate a very lightweight text editor: *Notepad--*. The following table shows all the effective keys in this editor and their effects. All other keys are ineffective and **should be ignored** by the editor.

`a-z`	Overwrite the selected segment with the character; if nothing is selected, simply write the character. Exit *selection mode*.
`H`	Move the cursor leftward by one. If the cursor is at the beginning file, do nothing.
`L`	Move the cursor rightward by one. If the cursor is at the end file, do nothing.
`V`	Enter *selection mode*.
`D`	Cut the selected segment and overwrite the clipboard with it. Exit *selection mode*.
`P`	Overwrite the selected segment with the content in the clipboard; if nothing is selected, simply paste the clipboard content then clear the clipboard. Move the cursor to the end of the pasted content. Exit *selection mode*.

Note that the selection mode works just like keeping <shift> key pressed in Notepad or Word. More precisely, in *selection mode*, the "selected segment" is the segment between the current cursor and the cursor position when you enter *selection mode*. While not in selection mode, nothing is selected.

Suppose *Arvin* just open an empty file with *Notepad--*, and the clipboard is initially empty. Given the series of keys pressed by *Arvin*, what would the file eventually be?

Input Format

The first line is an integer T, indicating that there are T test cases. For every test cases there is a string composed of lowercase and uppercase characters, which is the series of keys Arvin keys in.

Output Format

For every test case, output the final content of the file.

Sample Input

```
2
tleVHHHwaVHHac
```

```
dsahahaVHHHHDggHHHHHPLLLLLgggqaq
```

Sample Output

```
ac
hahadsaggggggqaq
```

Problem 5. The Crafty TA

Given 2 positive integers N, k, you can obtain a new integer by removing k digits from N. Please output the least one that does not contain any leading zeros.

Input Format

2 integers, N and k.

$1 \leq N \leq 10^{10^8}$

$1 \leq k \leq \min(\log_{10} n, 10^3)$

Output Format

An $(\log_{10} n - k)$-digit number, which is the least number you can obtain by removing k digits from N.

Sample Input

```
30400 2
```

Sample Output

```
300
```