

# Homework 1

● Graded

Student

苏慧哲

Total Points

48 / 48 pts

Question 1

Question 1 Academic Integrity

10 / 10 pts

1.1 a 2 / 2 pts

✓ - 0 pts Correct

- 2 pts Incorrect

1.2 b 2 / 2 pts

✓ - 0 pts Correct

- 2 pts Incorrect

1.3 c 2 / 2 pts

✓ - 0 pts Correct

- 2 pts Incorrect

1.4 d 2 / 2 pts

✓ - 0 pts A: Correct

- 1 pt AB: There is only one student A

- 2 pts Incorrect

1.5 e 2 / 2 pts

✓ - 0 pts Correct

- 2 pts Incorrect

Question 2

Question 2

2 / 2 pts

✓ - 0 pts Correct

- 2 pts Wrong

Question 3

Question 3

2 / 2 pts

✓ - 0 pts Correct

- 2 pts Wrong

Question 4

Question 4

2 / 2 pts

✓ - 0 pts Correct

- 2 pts Wrong

Question 5

Question 5 Array for List

10 / 10 pts

5.1 (a)

5 / 5 pts

✓ - 0 pts Correct

- 1 pt one wrong item

- 5 pts not selecting page

5.2 (b)

5 / 5 pts

✓ - 0 pts Correct

- 1 pt 1 mistake

- 4 pts 4 mistakes

- 5 pts not selecting page

- 1 pt 1 4 5 1 2 3 ..?

## Question 6

### Question 6 Postfix Expression

8 / 8 pts

6.1 (a)

3 / 3 pts

✓ - 0 pts Correct

- 1 pt in-fix expression wrong.
- 1 pt calculation procedure wrong.
- 2 pts Wrong change of stack.
- 0.5 pts Small mistake.
- 3 pts Please match all your pages correctly.

6.2 (b)

3 / 3 pts

✓ - 0 pts Correct

- 0.5 pts 1) wrong
- 0.5 pts 2) wrong
- 1 pt 3) wrong
- 1 pt 4) wrong
- 3 pts Please match the pages and questions correctly.

6.3 (c)

2 / 2 pts

✓ - 0 pts Correct

- 2 pts Wrong.

### Question 7

#### Question 7 Asymptotic Analysis

9 / 9 pts

7.1 — 1

3 / 3 pts

✓ - 0 pts Correct

- 3 pts Wrong answer.

- 1.5 pts Correct answer but wrong proof.

7.2 — 2

3 / 3 pts

✓ - 0 pts Correct

- 3 pts Wrong answer

- 1.5 pts Correct answer but wrong proof.

7.3 — 3

3 / 3 pts

✓ - 0 pts Correct

- 3 pts Wrong answer

- 1.5 pts Correct answer but wrong proof. (or lack proof)

### Question 8

#### Question 8 Complexity Analysis

5 / 5 pts

8.1 — 1

2 / 2 pts

✓ - 0 pts Correct

- 1 pt Partial Credit, without simplification or justification.

- 2 pts Wrong or Missing

8.2 — 2

3 / 3 pts

✓ - 0 pts Correct

- 1 pt Correct answer but wrong (or incomplete) proof.

- 3 pts Wrong Answer or Missing.

- 1 pt no final simplified answer

No questions assigned to the following page.

CS101 Algorithms and Data Structures  
Fall 2021  
Homework 1

---

Due date: 23:59, September 26, 2021

1. Please write your solutions in English.
2. Submit your solutions to [gradescope.com](https://gradescope.com).
3. Set your FULL Name to your Chinese name and your STUDENT ID correctly in Account Settings.
4. If you want to submit a handwritten version, scan it clearly. CamScanner is recommended.
5. When submitting, match your solutions to the according problem numbers correctly.
6. No late submission will be accepted.
7. Violations to any of the above may result in zero grade.

Questions assigned to the following page: [1.1](#), [1.2](#), [1.3](#), [1.4](#), and [1.5](#)

---

**1: (5\*2') Academic integrity**

---

**Question 1.** Please determine who violated academic integrity (or committed plagiarism) in the following situations. A stands for Alice, B stands for Bob, N stands for None, AB stands for Both.

Note that you should write you answers of section 1 in the table below.

Q1 (a)	Q1 (b)	Q1 (c)	Q1 (d)	Q1 (e)
AB	N	AB	A	AB

- (a) Alice and Bob are good friends, Alice asked if Bob could send her some code snippets so that she could have some ideas of what is going on in this homework (promising, of course, not copying his code). Bob agreed and sent his code repository to Alice, Alice copied the code and submitted to the CS 101 Platform.
- (b) Bob is Alice's boyfriend. In order to enhance their romantic relationship, they studied together in the ShanghaiTech library and collaborated on an individual programming assignment. They discussed about the algorithm using the whiteboard but they didn't see any lines of code from each other.
- (c) Alice and Bob are roommates. Bob let Alice use his laptop to play APEX Legends, and Alice copied Bob's solution for a paper assignment.
- (d) Alice found the solution for a specific assignment on github and copied it to her homework with some trivial modifications.
- (e) Alice borrows Bob's computer to complete her programming assignment. Without looking at Bob's code, she accidentally submits her own assignment using Bob's account to the CS 101 Platform. After that, Alice and Bob resubmitted their assignments separately using their own accounts.



Questions assigned to the following page: [2](#), [3](#), and [4](#)

---

**2: (3\*2') Multiple Choices**

---

Each question has one or more correct answer(s). Select all the correct answer(s). For each question, you get 0 point if you select one or more wrong answers, but you get 1 point if you select a non-empty subset of the correct answers.

*Note that you should write you answers of section 2 in the table below.*

Question 2	Question 3	Question 4
A	D	A

**Question 2.** Using a single linked list to implement a stack, suppose you only keep the head of it, where should the pushes and pops be performed in the optimal way?

- (A) Push in front of the first element, pop the first element
- (B) Push in front of the first element, pop the last element
- (C) Push after the first element, pop the first element
- (D) Push after the last element, pop the last element
- (E) Push after the last element, pop the first element

**Question 3.** Suppose we use a circular array with index range from 0 to  $N - 1$  to implement a queue, and currently Front is pointing at index  $m$ . We will know that if this queue is full the last element in it should be stored at index = \_\_\_\_\_. (Options below are mapped to **Integers Modulo  $N$** :  $\mathbb{Z}_N = \{0, 1, 2, \dots, N - 1\}$ )

- (A) 0
- (B)  $m$
- (C)  $N - 1$
- (D)  $m - 1$

**Question 4.** What function does the following code achieve?

```
void Q4(Queue &Q)
{
    Stack S;
    int d;
    S.InitStack();
    while(!Q.IsEmpty())
    {
        d = Q.DeQueue();
        S.Push(d);
    }
    while(!S.IsEmpty())
    {
        d = S.Pop();
```

Question assigned to the following page: [4](#)

```
        Q.Enqueue(d);  
    }  
}
```

- (A) Use a stack to reverse a queue.
- (B) Use a queue to reverse a stack.
- (C) Use a stack to implement a queue.
- (D) Use a queue to implement a stack.

Questions assigned to the following page: [5.1](#) and [5.2](#)

---

**3: (5'+5')Array for list**

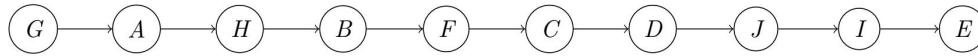

---

**Question 5.** In this question, we use two arrays: “next” and “value” to simulate a linked-list. The elements at each index in “next” and “value” are combined together to represent a node in the linked-list, where “next” represents the pointer to the next node, and “value” represents data in the nodes.

(a) (5') **Linked list from array representation**

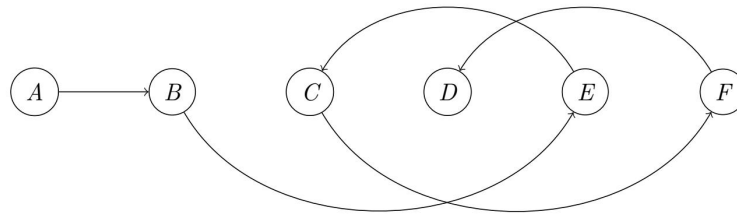
Here are the two arrays of “next” and “value”. Please draw the linked list presented by the two arrays.

<i>index</i>	0	1	2	3	4	5	6	7	8	9
<i>value</i>	A	B	C	D	E	F	G	H	I	J
<i>next</i>	7	5	3	9	/	2	0	1	4	8



(b) (5') **Array representation for a linked list**

Fill in the “next” array in order to make the array and the linked-list showing below equivalent.



<i>index</i>	0	1	2	3	4	5
<i>value</i>	A	B	C	D	E	F
<i>next</i>	1	4	5	/	2	3

Questions assigned to the following page: [6.1](#), [6.2](#), and [6.3](#)

---

**4: (8') Postfix expression**


---

**Question 6.** Reverse Polish notation (RPN) is a mathematical notation in which operators follow their operands. Using a stack, we can evaluate postfix notation equations easily.

**(a) (1'+2') Calculating**

A post-fix expression (Reverse-Polish Notation) with single digit operands is shown below: (where  $^$  is the exponentiation operator)

$$8\ 2\ 3\ ^\wedge / 2\ 3\ * + 5\ 1\ * -$$

Its in-fix expression is:  $8/(2^3) + (2*3) - (5*1)$

The changing of the stack to calculate the final result is:

		3				3				1		
	2	2	8		2	2	6		5	5	5	
8	8	8	8	1	1	1	1	7	7	7	7	2

**(b) (2\*0.5'+2\*1') Conversion**

Now, try to convert the following in-fix expression into post-fix expression: (You don't need to calculate them)

1)  $1 + 2 + 3$

$$1\ 2\ +\ 3\ +$$

2)  $1 * 2 + 3$

$$1\ 2\ *\ 3\ +$$

3)  $1 + 2 * 3 + (4 * 5 + 6) * 7$

$$1\ 2\ 3\ *\ +\ 4\ 5\ *\ 6\ +\ 7\ *\ +$$

4)  $1 + (2 * 3^4) / (5 + 6) + 7$

$$1\ 2\ 3\ 4\ ^\wedge *\ 5\ 6\ +\ /\ +\ 7\ +$$

**(c) (4\*0.5') Validity**

Please judge whether the following post-fix expression is legal, if legal, please write **T**, otherwise please write **F**.



Question assigned to the following page: [6.3](#)

$$(a) \quad 1 \ 2 \ * \ + \ 3 \ 5 \ +$$

***F***

$$(b) \quad 4 \ 5 \ 6 \ / \ * \ 1 \ /$$

***T***

$$(c) \quad 1 \ + \ 2 \ - \ 3 \ + \ 4$$

***F***

$$(d) \quad 7 \ 8 \ 9 \ 1 \ + \ - \ *$$

***T***

Questions assigned to the following page: [7.1](#) and [7.2](#)

---

**5: (9'+5') Big-O Notation**


---

**Question 7. (3\*3') Asymptotic Analysis**

For each pair of functions  $f(n)$  and  $g(n)$ , give your answer whether  $f(n) = O(g(n))$ ,  $f(n) = \Omega(g(n))$  or  $f(n) = \Theta(g(n))$ . Give a **proof** of your answers.

Note 1: Try to give your answer in the most precise form. For example, for  $f(n) = n^2$  and  $g(n) = n^2 + 2n + 1$ , write  $f(n) = \Theta(g(n))$ .

Note 2: Try to prove by calculating limits.

1.  $f(n) = 1.01^n$  and  $g(n) = n^{10}$

**Answer:**  $f(n) = \Omega(g(n))$

**Proof:**

By L'Hospital's rule, we can calculate the limits of  $\frac{f(x)}{g(x)}$

$$\begin{aligned}
 \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} &= \lim_{n \rightarrow \infty} \frac{(1.01)^n}{n^{10}} \\
 &= \lim_{n \rightarrow \infty} \frac{\frac{1}{1.01} \cdot (1.01)^n}{10 \cdot n^9} && \text{(L'Hospital's rule)} \\
 &= \dots = \lim_{n \rightarrow \infty} \frac{(\frac{1}{1.01})^{10} \cdot (1.01)^n}{10!} && \text{(L'Hospital's rule)} \\
 &= \frac{1}{(1.01)^{10} \cdot 10!} \cdot \lim_{n \rightarrow \infty} (1.01)^n
 \end{aligned}$$

When  $n \rightarrow \infty$ ,  $\frac{f(n)}{g(n)} \rightarrow \infty$ .

Therefore,  $f(n) = \Omega(g(n))$

2.  $f(n) = \log(n!)$  and  $g(n) = \log(n^n)$

**Answer:**  $f(n) = \Theta(g(n))$

**Proof:**

$$\begin{aligned}
 f(n) &= \log(n!) \\
 &= \log(1 \cdot 2 \cdot 3 \cdots n - 1 \cdot n) \\
 &= \log 1 + \log 2 + \dots + \log(n-1) + \log(n) \\
 &< \log(n) + \log(n) + \dots + \log(n) \\
 &= n \log(n) = \log(n^n) \\
 &= g(n)
 \end{aligned}$$

Therefore,

$$\lim_{n \rightarrow \infty} \frac{f(x)}{g(x)} \leq 1$$

We also know that at least half of the term

$$\sum_{k=1}^n \log k$$

Questions assigned to the following page: [7.2](#) and [7.3](#)

is greater than  $\log\left(\frac{n}{2}\right)$ . Thus, we have

$$f(n) \geq \frac{n}{2} \log\left(\frac{n}{2}\right)$$

Using L'hospital's rule again, we can calculate the lower bound of the limits.

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} &\geq \lim_{n \rightarrow \infty} \frac{\frac{n}{2} \log\left(\frac{n}{2}\right)}{n \log n} \\ &= \frac{1}{2} \lim_{n \rightarrow \infty} \frac{\log\left(\frac{n}{2}\right)}{\log n} \\ &= \frac{1}{2} \lim_{n \rightarrow \infty} \frac{\frac{2}{n} \cdot \frac{1}{2}}{\frac{1}{n}} && \text{(L'Hospital's rule)} \\ &= \frac{1}{2} \end{aligned}$$

Then we can conclude that there  $\exists c, \frac{1}{2} \leq c \leq 1$ , such that

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c$$

Thus,  $f(n) = \Theta(g(n))$

3.  $f(n) = \log^2(n)$  and  $g(n) = \log \log n$

**Answer:**  $f(n) = \Omega(g(n))$

**Proof:**

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} &= \lim_{n \rightarrow \infty} \frac{\log^2(n)}{\log \log n} \\ &= \lim_{n \rightarrow \infty} \frac{2 \log n \cdot \frac{1}{n}}{\frac{1}{\log n} \cdot \frac{1}{n}} && \text{(L'Hospital's rule)} \\ &= \lim_{n \rightarrow \infty} 2 \log^2(n) \end{aligned}$$

When  $n \rightarrow \infty$ ,  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \rightarrow \infty$ .

Therefore,  $f(n) = \Omega(g(n))$

Questions assigned to the following page: [8.1](#) and [8.2](#)

**Question 8. (5') Complexity Analysis**

Consider the factorial computing problem:  $N! = 1 \times 2 \times \dots \times N$ .

1. (2') To store the number  $N$ , we need at least  $\log N$  bits of space. Find an  $f(N)$  such that  $N!$  is  $\Theta(f(N))$  bits long. Simplify your answer and justify your answer.

$$f(N) = N \log N$$

We have already proved that

$$N \log N = \Theta(\log(N!))$$

in question 7.2.

We also know that  $N!$  is at least  $\lfloor \log_2 N \rfloor + 1$  bits long, and at most  $\lceil \log_2 N \rceil + 1$  bits long. Therefore,  $N!$  must be  $\Theta(\log(N!)) = \Theta(N \log N) = \Theta(f(N))$  bits long.

2. (3') Consider this naive factorial computing algorithm.

```

procedure FACTORIAL( $N$ )
  if  $N > 1$  then
    return FACTORIAL( $N - 1$ )  $\cdot N$ 
  else
    return 1
  end if
end procedure

```

We assume the runtime of multiplying an  $m$ -bit number and an  $n$ -bit number is  $\Theta(mn)$ .

What's the runtime of this factorial computing algorithm in Big- $O$  notation? Justify your answers.

Let  $T(N)$  be the complexity of the algorithm when input  $N$

$$\begin{aligned}
 T(n) &= T(n-1) + \Theta((n-1) \cdot \log(n-1) \cdot \log n) \\
 T(0) &= \Theta(1) \\
 T(N) &= \Theta\left(2 + \sum_{k=2}^N (N-1) \cdot \log N \cdot \log(N-1)\right) \\
 &= \Theta\left(\sum_{k=2}^N (N-1) \cdot \log N \cdot \log(N-1)\right) \\
 &= O\left(\sum_{k=2}^N N \cdot \log^2 N\right) \\
 &= O((N-2) \cdot N \cdot \log^2 N) \\
 &= O(N^2 \cdot \log^2 N)
 \end{aligned}$$