CS 101 Fall 2021 - Quiz 1-A

9/27/2021 - 25 Minutes **Name:** **ID number:** **Score:**

Remember that your work is graded on the quality of your writing and explanation as well as the validity.

Problem 1 (5pts) Notes of discussion

I promise that I will complete this QUIZ independently, and will not use any electronic products or paper-based materials during the QUIZ, nor will I communicate with other students during this QUIZ.

True or False: I have read the notes and understood them.

1
T

Problem 2(10pts) Stack and Queue

(1) (6 Points) Suppose there is an initially empty stack with the capacity 7, then we do a sequential of 7 `push` and 7 `pop` operations. If the order of the element pushed in the stack is 1 2 3 4 5 6 7, then for each order of the popped elements listed below, tick a "✓" in the box if it could be existing.

7 6 5 4 3 2 1	✓
2 4 6 7 5 3 1	✓
1 3 4 6 7 5 2	✓
2 1 4 5 3 6 7	✓
1 5 4 3 2 6 7	✓
4 5 3 6 2 7 1	✓

(2) (4 Points) Suppose there is an initially empty queue with capacity 7 which is implemented by an array (viewed circularly). Show the array after the following operations being operated and indicate the place of the front and back of the queue.

(a) Enqueue(1) Enqueue(3) Enqueue(5)
Dequeue()
Enqueue(7) Enqueue(9) Enqueue(1)
Dequeue()
Enqueue(3) Enqueue(5) Enqueue(7)
Dequeue()
5 7(B) □ 7(F) 9 1 3

(b) Enqueue(1) Enqueue(2) Enqueue(7) Enqueue(6) Enqueue(5)
Dequeue()
Enqueue(1) Enqueue(3) Enqueue(5)
Dequeue() Dequeue() Dequeue()
Enqueue(6) Enqueue(7) Enqueue(8)
Dequeue() Dequeue()
5 6 7 8(B) □ □ 3(F)

CS 101 Fall 2021 - Quiz 1-A

9/27/2021 - 25 Minutes **Name:** **ID number:** **Score:**

Problem 3(10pts) Algorithm Design

(1) (6 Points) Try to convert the polynomial below into the array form which is talked in the class. Note the exponents should be descending.

$$2200x^{2800} + 4396x^{777} + 443x$$

index	0	1	2
coefficient	2200	4396	443
exponent	2800	777	1

(2) (4 Points) Try to do addition on the two polynomial A and B below and store the result in C. Each polynomial is stored in the struct `PLY`.

```
struct PLY {
    int exponent[VERY_LARGE];
    int coefficient[VERY_LARGE];
    int len;
};

PLY add(PLY &A, PLY &B) {
    PLY C;
    int i = 0;
    int j = 0;
    int k = 0;
    while (__i < A.len__ or j < B.len) {
        if (j >= B.len or i < A.len and A.exponent[i] > B.exponent[j]) {
            C.exponent[k] = A.exponent[i];
            C.coefficient[k] = A.coefficient[i];
            k++;
            i++;
        } else if (__i >= A.len__ or __j < B.len__ and A.exponent[i] < B.exponent[j
            ]) {
            C.exponent[k] = B.exponent[j];
            C.coefficient[k] = B.coefficient[j];
            k++;
            j++;
        } else if (A.exponent[i] == B.exponent[j]) {
            C.exponent[k] = A.exponent[i];
            C.coefficient[k] = __A.coefficient[i] + B.coefficient[j]__;
            k++;
            i++;
            j++;
        }
    }
    C.len = k;
    return C;
}
```

CS 101 Fall 2021 - Quiz 1-A

9/27/2021 - 25 Minutes **Name:** **ID number:** **Score:**

Problem 4(16pts) Asymptotic Analysis

(1) (10') Order the following functions so that for all i, j, if f_i comes before f_j in the order then $f_i = O(f_j)$. Do **NOT** justify your answers.

- $f_1(n) = \sqrt{n}$
- $f_2(n) = n^{\frac{1}{4}}$
- $f_3(n) = 5000$
- $f_4(n) = 2^{\log_2 n}$
- $f_5(n) = 3^n$
- $f_6(n) = \frac{1}{2}^n$
- $f_7(n) = \log_2 n$
- $f_8(n) = 2^{\sqrt{n}}$
- $f_9(n) = 3^{\log_2 n}$
- $f_{10}(n) = n!$

As an answer you may just write the functions as a list, e.g. $f_8, f_9, f_1, \cdots$

$f_6, f_3, f_7, f_2, f_1, f_4, f_9, f_8, f_5, f_{10}$

Note: Polynomial dominates Logarithm, Exponential dominates Polynomial.

(2) (6') 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.

- $f(n) = e^n$ and $g(n) = n^\epsilon, \forall \epsilon > 0$

  $f(n) = \omega(g(n))$

  -Using L'Hospital's rule:

$$\lim_{x\to\infty} \frac{e^x}{x^\epsilon} = \lim_{x\to\infty} \frac{\frac{d}{dx}e^x}{\frac{d}{dx}x^\epsilon} = \lim_{x\to\infty} \frac{e^x}{\epsilon x^{\epsilon-1}} = ... = \infty$$

  Therefore $e^n = \omega(n^\epsilon)$

- $f(n) = n!$ and $g(n) = n^n$

  $f(n) = o(g(n))$

  - One way to prove: Prove by the limit condition:

$$\lim_{n\to\infty} \frac{n!}{n^n} = 0$$

  which can be proved by the following statements: we have $n \geq 2k$ for every $1 \leq k \leq n/2$ and $n \geq k$ for every $n/2 < k \leq n$, hence

$$n^n = \prod_{k=1}^{n} n \geq \prod_{1\leq k\leq n/2} (2k) \cdot \prod_{n/2<k} k = 2^{n/2} \cdot n!$$

  then we have,

$$\lim_{n\to\infty} \frac{n!}{n^n} \leq \lim_{n\to\infty} \frac{1}{2^{n/2}} = 0$$

  - Another way to prove: First prove $n! = O(n^{n-1})$ (which can be easily proved by definition of finding $c = 1, N = 1, \forall n \in \mathbb{N} \geq N, n! \leq c \cdot n^{n-1}$), then prove that $n^{n-1} = o(n^n)$, therefore, $n! = o(n^n)$