General instruction: This assignment includes two parts, written and programming. Please write/type your answers neatly so they can be readable. Please submit a single PDF file for the written part and a zip file for the programming part before **Sunday**, **11:59 PM Sept. 25th, 2022**.

Special office hour for this assignment will be on Wednesday, Sept. 21th, 2022, 6:00 pm-7:00 pm via Zoom (meeting ID: 273 935 2895, passcode: 1) or Olin 503. You can also send me an email with your questions at wxl387@case.edu.

Written Problems (50 pts)
(*Show steps for partial credits*)

1. Determine whether the statement is true or false. (4 pts)
   (a) 2^n grows faster than n^3
   (b) $2^{2n} = O(2^n)$

2. Please simplify the following Big-O notation. (6 pts)
   (a) $O(2n^4 + 6n + 10000)$
   (b) $O(log_2 n^2 + (log_2 n)^2 + log_2 n)$
   (c) $O((n+3)^4 + (n+5)^2)$

3. Provide a tight Big-O notation for each of the code snippets. (40 pts)
   (a)

```
public int calcSum1(int n) {
        int sum = 0;
        for (int i = 1; i < n; i *= 2) {
                sum++;
        }
        return sum;
}
```

   (b)

```
public int calcSum2(int n) {
        int sum = 0;
        for (int i = 1; i < n; i++) {
                for (int j = 1; j < n/i; j++) {
                        sum++;
                }
        }
        return sum;
}
```

   (c)

```
public int calcSum3(int n) {
	int sum = 0;
	for (int i = n; i > 1; i--) {
		for (int j = i; j > 1; j=j/3) {
			sum++;
		}
	}
	return sum;
}
```

(d)

```
public int calcSum4(int n) {
	int sum = 0;
	for (int i = 0; i < n; i++) {
		for (int j = 1; j < i*i; j++) {
			if(j % i == 0) {
				for(int k = 0; k < j; k++) {
					sum++;
				}
			}
		}
	}
	return sum;
}
```

Programming Problems (50 pts)

1. Write a program in both iterative and recursive approaches that prints out the n^{th} Fibonacci element(not the whole sequence). Name the method as you like(e.g., fibonacciIteration(int n) and fibonacciRecursion(int n)). Please consider the efficiency of the two methods in terms of Time Complexity and type the Big-O notation down as comments. Call the method that is more efficient in the main method. (15 pts)

2. Based on the program you created, further develop the program that adds each Fibonacci number(starting from 0, 1) to a list(an array), and can perform remove, ifContains, grab, and numItems operations. (35 pts)

- add(int item): add a number to the list (allow to add duplicates)
- remove(int item): remove the item (if exists) from the list
- ifContains(int item): check if the item exists in the list
- grab(): random draw a number from the list without removing
- numItems(): prints out the number of unique items in the list (without duplicates)

Grading:

- Written problems
    - 100% if the final answer is correct,
    - Explanations or thoughts are suggested, will offer partial credit if the final answer is incorrect.
- Programming problems
    - Mainly based on correctness, slightly on efficiency,
    - Please write down any additional resources other than lecture slides as comments,
    - Please also list any collaborators as comments.