

C S 272/463 Introduction to data structures

Fall 2019

Lab 10: Recursive thinking

1 Learning objectives

Objective 2 (recursive thinking), Objective 5, Objective 6, and Objective 7.

2 Requirements

2.1 Task

- Implement and test recursive algorithms described below. With the proper use of recursion, none of these methods should require more than a dozen lines of code.
Please put all the following function to a file **RecursiveQuestion.java**.

2.2 Detailed instructions for program design and implementation

1. (10 points) The Fibonacci numbers are recursively defined as follows:

```
F_0 = 0
F_1 = 1
F_n = F_(n-1) + F_(n-2) for n > 1
```

Please write an algorithm to calculate the k th Fibonacci number F_k using binary recursion.

```
public static int FibBinaryRecursive(int k)
```

2. (15 points) **HanoiTower** function.

Write an algorithm to solve the Towers of Hanoi problem. (You can take Project 11 at page 440 in your text book as reference).

3. (15 points) Write a method

```
public static String showCallLevel(int L, int curl)
```

which produces the output shown below.

```
This was written by call number 1x.
  This was written by call number 2x.
    This was written by call number 3x.
      This was written by call number 4x.
        This was written by call number 4y.
          This was written by call number 3y.
            This was written by call number 2y.
              This was written by call number 1y.
```

The above example is the output of `showCallLevel(4,1)`, its recursion stopped when it reached four levels deep. But your method should be capable of continuing to any specified level L .

4. (20) **BinaryPrint** function. Write a Java method with the following header

```
public static void BinaryPrint (int n)
```

The number `n` is non-negative. The method prints the value of `n` as a binary number. If `n` is zero, then a single zero is printed; otherwise, no leading zeros are printed in the output. Your implementation must be recursive.

Examples:

`n=0` Output: 0

`n=4` Output: 100

`n=27` Output: 11011

5. (20 points) **Pattern** function Write an algorithm to solve the programming project 9 at page 440 of your text book.
6. (20 points) **Permutation.java**.
You are given an array of distinct integers, you are required to write a recursive method that prints the permutations of the integers in this array.
Write a test function to test your Permutation algorithm. Test your algorithm when the array length is 1, 2, 5, and 10.

3 Note

- **Specifications** for all your classes and methods:
Please properly explain (1) the functionality of the methods, (2) the parameters, (3) the return values, (4) the pre-conditions if there is any;
Please use inline comments, meaningful variable names, indentation, formatting, and whitespace throughout your program to improve its readability.
- You can (but are not required to) design and implement other facilitating methods (E.g., other get and set methods, toString method) to finish the implementation of the required methods.

4 Submission

Submit through canvas a zipped file containing your java file(s) (not `.class` files).

5 Grading criteria

- (1) The score allocation is already put in the questions.
- (2) Please make sure that you test your code **thoroughly** by considering all possible test cases.
- (3) 5 points will be deducted if submitted files (including files types, file names, etc.) do not follow the instructions.
- (4) At least 20 points will be deducted if your code cannot be run on CS servers.