

CISC 3115 EWQ6

# Recursion and Strings

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

# Outline

- Discussed
  - Problem Solving using Recursion
  - Recursive math functions
  - Design solutions to recursive math functions using recursion
- Recursions and Strings
  - Print messages n times
  - Is the string a palindrome?

# Characteristics of Recursion

- All recursive methods have the following characteristics:
  - One or more base cases (the simplest case) are used to stop recursion.
  - Every recursive call reduces the original problem, bringing it increasingly closer to a base case until it becomes that case.

# Recursion as Problem Solving Strategy

- Break the problem into subproblems such that one or more subproblems resembles the original problem
  - These subproblems resembling the original problem is almost the same as the original problem in nature with a smaller size.
- Apply the same approach to solve the subproblem recursively to reach the base case

# Print Message Many Times

- Problem: print a message  $n$  times.
- Can we solve it using recursion?
- Original problem: print a message  $n$  times
- Subproblems with smaller size
  - print a message 1 time
  - print a message  $(n-1)$  times (the same problem as the original problem, but smaller size)
- Base case: print the message 0 times (or 1 time)

# Print Message Many Times: Solution

- Base case:  $n = 0$  (how about the  $n = 1$  base case? Which one yields better code?)

```
public class PrintMsg {  
    public static void main(String[] args) {  
        nPrintMsg("Hello, World!", 5);  
    }  
  
    public static void nPrintMsg(String msg, int n) {  
        if (n == 0) return; // base case  
        System.out.println(msg); // subproblem 1  
        nPrintMsg(msg, n-1); // subproblem 2  
    }  
}
```

# Problem Solving Example: Print Message Many Times: Revisit

- Wait! Can we just write a loop to print a message  $n$  times? (solving problem iteratively)
- Remarks
  - However, it is sometimes easier to think recursively to solve a complex problems.
  - Many problems we solve iteratively can also be solved recursively.
    - Example: the palindrome problem (e.g., madam, nursesrun)

# Is It a Palindrome?

- Problem: is a given string a palindrome?
- Recursive solution:
  - 1) Compare the first and last character of the string. If not equal, not palindrome; 2) otherwise, repeat for the substring less the first and the last character (the same problem whose size is the original size – 2)
  - Base case: a single character or empty string, and the single character string or the empty string is always a palindrome.

# Is It a Palindrome? Solution

- An example realization of the solution

```
public static boolean isPalindrome(String s) {  
    // base case  
    if (s.length() <= 1) return true;  
  
    // subproblem 1  
    if (s.charAt(0) != s.charAt(s.length()-1)) return false;  
  
    // subproblem 2  
    return isPalindrome(s.substring(1, s.length()-1));  
}
```

# Questions

- Characteristics of recursion
- Recursion as problem solving strategy
- String operations and recursion