Recursion and Backtracking

Data Structures CS 218

Week 2 and Week 3

Introduction

- Basics of Recursion
- Different types of Recursion
  - Direct
  - Indirect
  - Tail
  - Non Tail

Recursion: The main ingredients

- To formulate a recursive solution:
  - Identify the "simplest" instance

    The **base case(s)** that can be solved *without* recursion
  - Identify "simpler" instances of the same problem

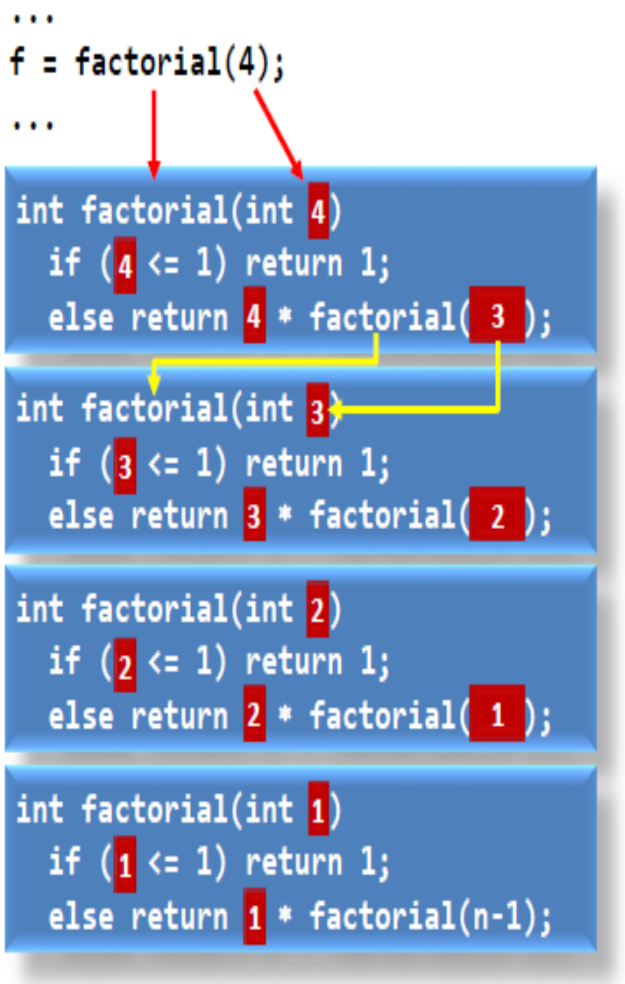
    The **recursive case(s)** that requires recursive calls to solve them
    - Identify how the solution from the simpler problem can help to construct the final result
  - Be sure we are able to reach the "simplest" instance
    - So that we will not get an **infinite recursion**

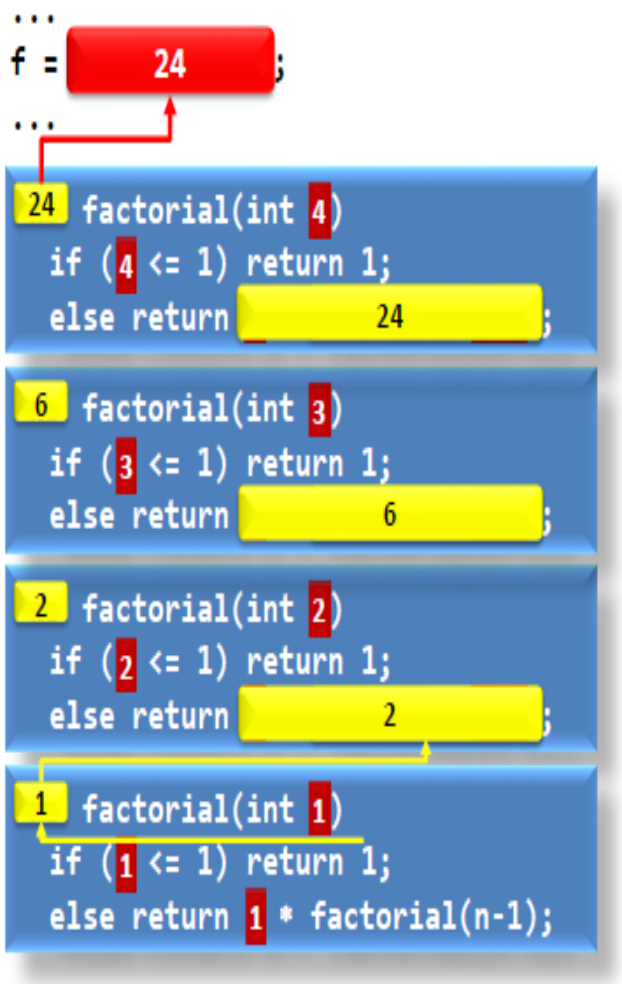
Examples of Recursion discussed in class

- Finding factorial
- Traversing linked list
- Reversing linked list
- Fibonacci series
- Sum of digits
- Digital Root
- Tower of hanoi
- Backtracking
  - N-Queen problem
  - Maze problem

Recursion Behind the Scenes/ Activation Records using Stacks

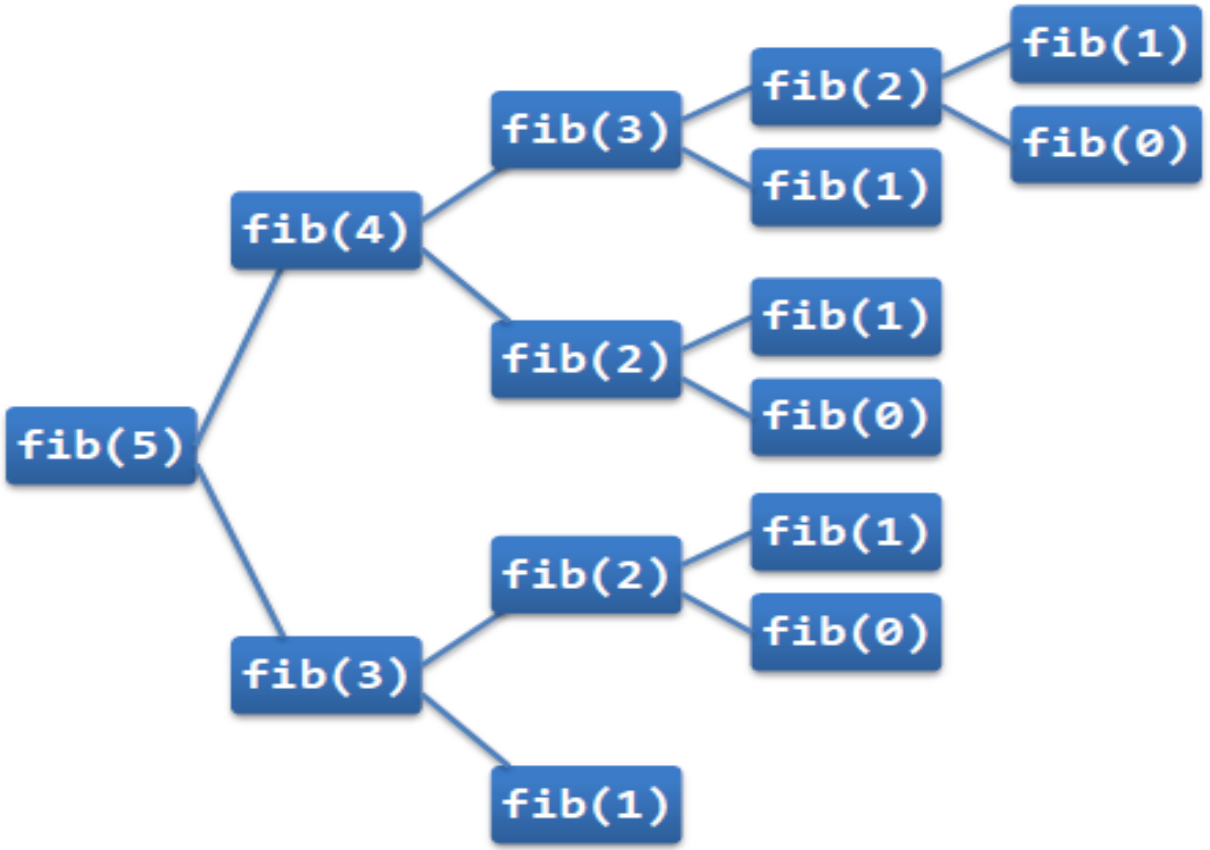
Finding Factorial



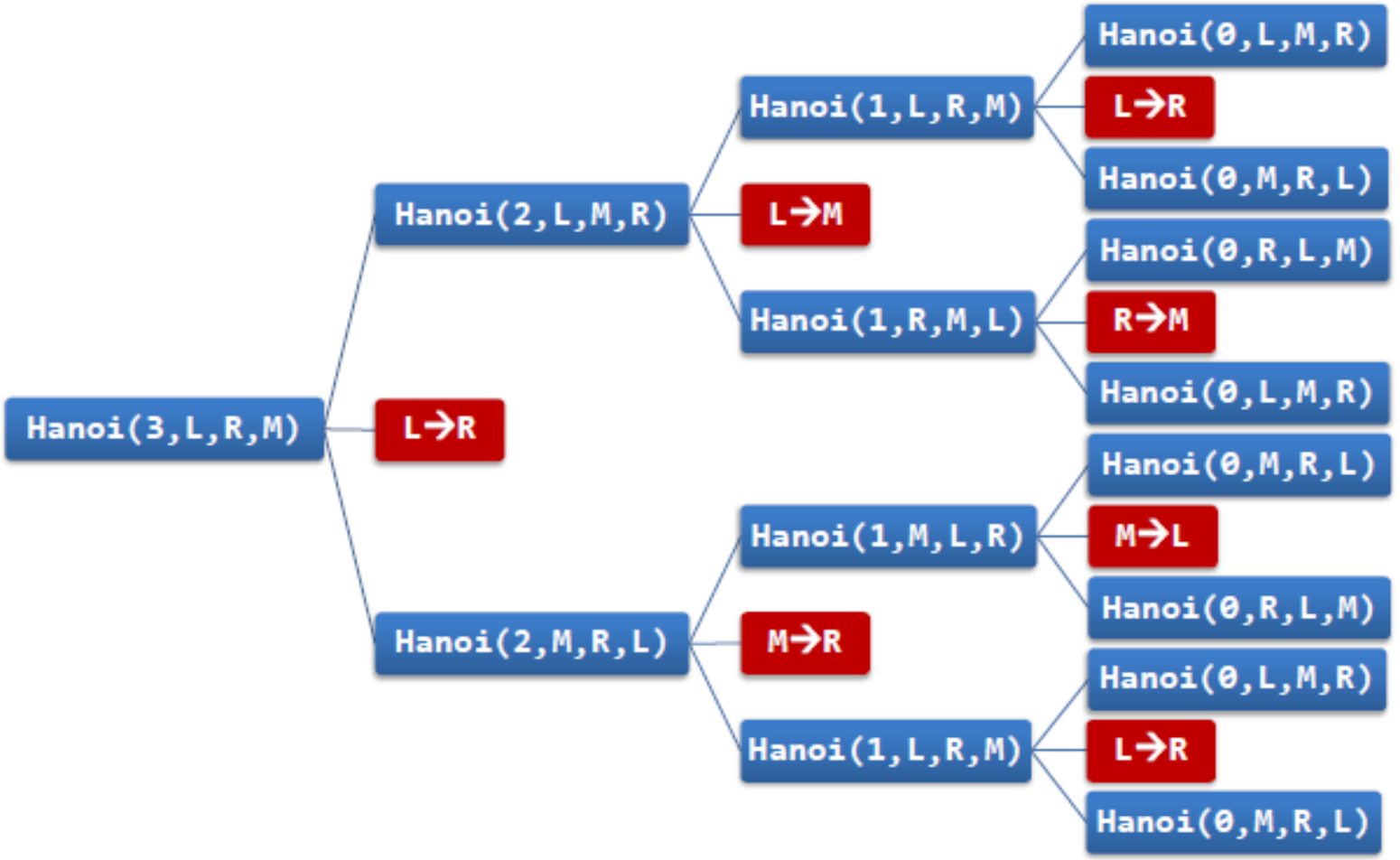


Fibonacci Series

The tree of calls for fib(5) would be:



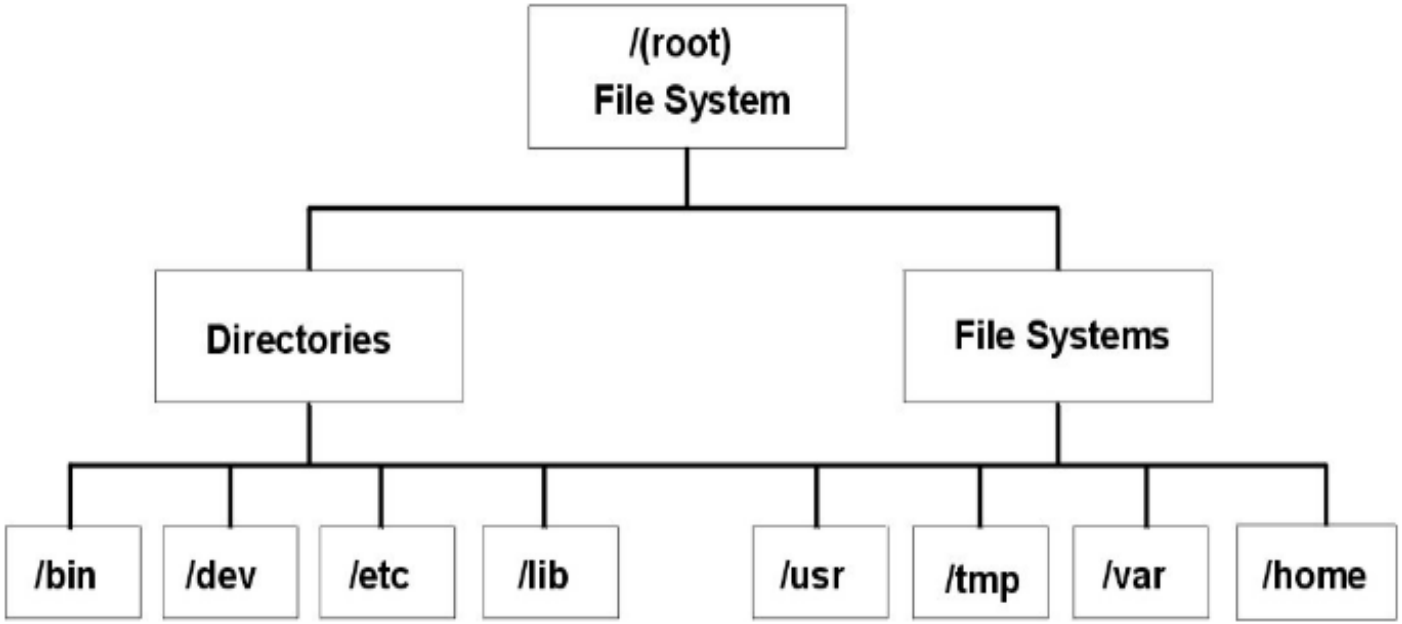
Tower of Hanoi



Iterative methods Vs Recursive methods

- Iterative methods vs recursion
- When to use iterative methods?
- When to use recursion?
- When not to use recursion?

An other good example



Backtracking

- Introduction to backtracking
- Is recursion and backtracking are same?
- N-Queen Problem discussion