Recursion

Chapter 16

S. Dandamudi

Outline

- Introduction
- Recursion
- Recursion vs. Iteration

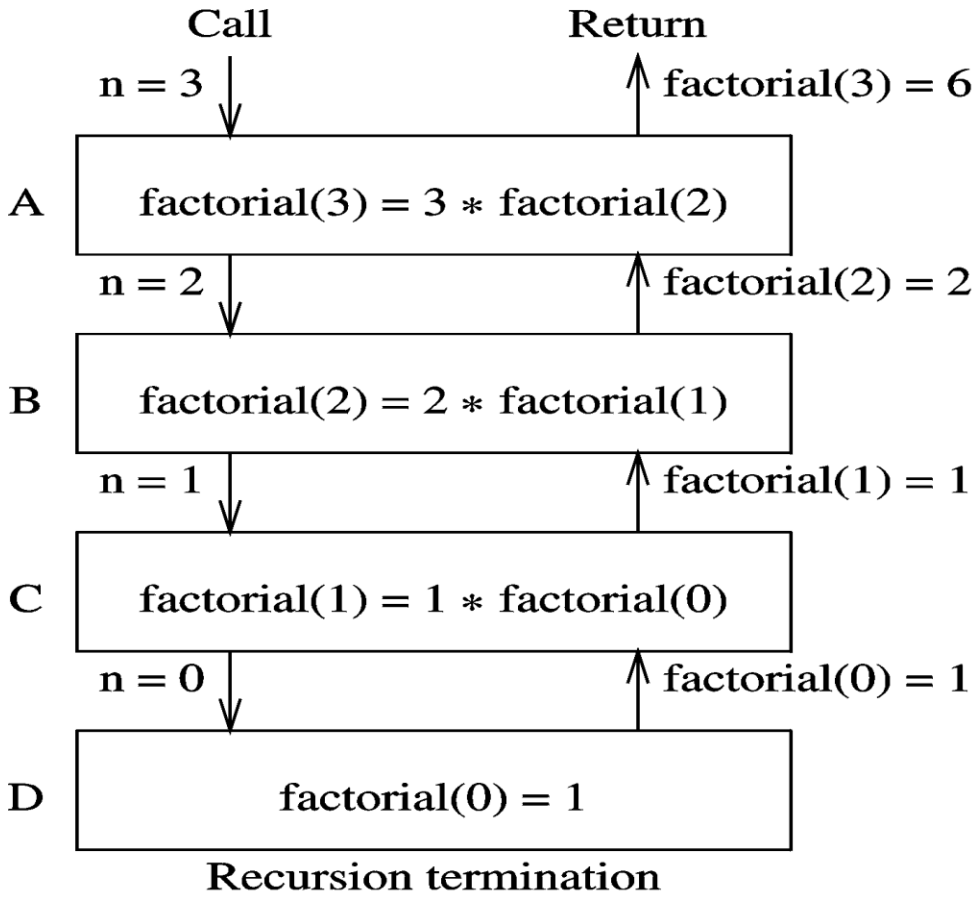
Introduction

- A recursive procedure calls itself
  - Directly, or
  - Indirectly
- Some applications can be naturally expressed using recursion

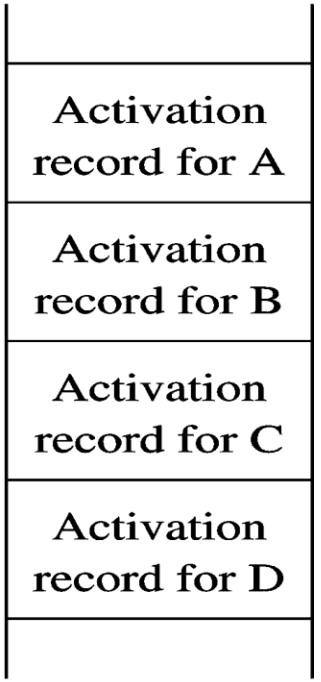
  factorial(0) = 1

  factorial (n) = n * factorial(n−1) for n > 0
- From implementation viewpoint
  - Very similar to any other procedure call
    - Activation records are stored on the stack

Introduction (cont'd)



(a)



(b)

Recursion

- Two examples in Pentium and MIPS assembly languages
  - Factorial
  - Quicksort

Example 1

- Factorial

  factorial(0) = 1

  factorial (n) = n * factorial(n–1) for n > 0

- Activation record
  - Consists of the return address pushed onto the stack by the call instruction

Recursion (cont'd)

Example 2

- Quicksort
  - Sort an N-element array
  - Basic algorithm
    - Selects a partition element ***x***
    - Assume that the final position of ***x*** is array[*i*]
    - Moves elements less than ***x*** into array[0]…array[*i*−1]
    - Moves elements greater than ***x*** into array[*i*+1]…array[*N*−1]
    - Applies quicksort recursively to sort these two subarrays

Recursion vs. Iteration

- Recursion
  - Concise
  - Better program maintenance
  - Natural choice for some problems
- Potential problems
  - Inefficiency
    - Call invocation and return overheads
    - Duplicate computation
  - Increased memory requirements