Comp 352 Winter 2019

Tutorial Week 3

Ling Tan[†]

January 21, 2019

[†]lingt.xyz@gmail.com

Outline

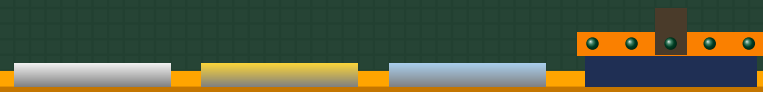
1. Recursion

2. Recursion Application Exercises

3. Complexity Analysis Problems

Definition

- Recursion occurs when a certain function calls itself in its own definition.
- Typically, a recursive function contains:
  - One or more base cases, which are not recursive. These work as a final goal or a stopping point.
  - One or more recursive cases, that solve a small part of the problem and then call the function again to solve the rest.



Recursion Types

- Linear Recursion:
  - Simplest form of recursion
  - Only one recursive call is made per function call
- Tail Recursion:
  - Same as Linear Recursion
  - Recursive call must be the last operation if called

Question

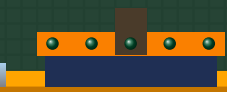
Does the following algorithm use tail recursion?

Algorithm 1 LinearSum(A, n)

Input: An integer array A and an integer $n \geq 1$, such that A has at least n elements

Output: The sum of the first n integers in A

```
1: if (n=1) then
2:     return A[0]
3: else
4:     return LinearSum(A, n − 1) + A[n − 1]
5: end if
```



Recursion Types

- Binary Recursion:
  - Recursive cases call the function two more times
- Multiple Recursion:
  - Generalization of binary recursion, with multiple recursive calls (more than 2 calls)

C3.7

Give a recursive algorithm to compute the product of two positive integers m and n, using only addition and subtraction.

Describe a recursive algorithm for finding the maximum element in an array A of n elements.

Given an array A of length n containing values in increasing order, write a recursive algorithm to find the first repeated pair of values if such a pair exists.

Complexity Analysis Problems

Consider the following code, n is data size, k is a constant

```
1: for (i = 0; i < n; i = i + k) do
2:     for (j = 0; j < i; j++) do
3:         sum[j] = j * sum[i];
4:     end for
5: end for
```

What is the big-O time complexity in terms of n? Show all necessary steps.

Prove that the running time

$$T(n) = n^3 + 20n + 1 \text{ is } O(n^4)$$

Prove that the running time

$$T(n) = n^3 + 20n + 1 \text{ is } \Omega(n^2)$$