

CSI 32 LECTURE NOTES (Ojakian)

Topic 10: Functional Recursion

OUTLINE

(References: 11.3, 11.4)

1. Functional Recursion
 2. Binary Search
-

1. Examples

PROBLEM 1. Write a function that computes the factorial. Do it in two ways: First with loops and then with functional recursion.

PROBLEM 2. Calculate the maximum of a list using recursion (without using the `max` function).

2. Functional Recursion

- (a) Must have sufficient base cases.
- (b) Must make “progress” towards base case
- (c) Strategy:
 - i. Calculates base case(s)
 - ii. Non base cases: Calculated correctly assuming the simpler case calculates correctly.
- (d)

***PROBLEM* 3.** Use recursion to find the sum of the squares of a list of numbers.

PROBLEM 4. Let $P(k, n) =$ number of unordered partitions of n into positive integers that are all at least k . Note the following recurrence:

$$P(k, n) = P(k + 1, n) + P(k, n - k)$$

- i. Calculate $P(2, 5)$ using the definition.
- ii. Determine base cases for the recurrence.
- iii. Calculate $P(2, 5)$ using the recurrence and the base cases.
- iv. Write a recursive program to calculate $P(k, n)$.
- v. Make a rough calculation of “how long” the program takes to execute.

3. Binary Search

PROBLEM 5.

- (a) In the worst case, how long does it take to search a list to determine whether or not it contains a given value?
- (b) Write a program that implements **Binary Search** to search a sorted list in order to determine if a given value is in the list or not.