

CS 367 - Introduction to Data Structures

Thursday, March 10, 2016

Program 3 assigned – FINISH IT BEFORE BREAK

Homework 6 assigned Monday

Last Time

Call Stack Tracing

Recursion

- recursion vs. iteration
- rules of recursion
- constructing recursive code
- practice writing recursive code

Exam 1 returned

Today

Recursion

- more practice writing recursive code
- complexity of recursive methods
- practice analyzing complexity

Next Time

Read: finish *Recursion*, *Search*

Read: start *Trees*

Execution tree tracing

Searching

Categorizing ADTs Part 1

General Trees

- implementing
- determining tree height

Practice – Array

→ Write a recursive method that counts the number of even values in an (non-null) array filled with integers.

1.

2.

3.

4.

```
int evenCount(int[] array) {
```

Analyzing Complexity of Recursive Methods

Options:

- 1.
- 2.

Steps

- 1.

- 2.

- 3.

- 4.

Practice – Complexity of Recursive `evenCount`

Problem size N is

1. Equations

2. Table

3. Verify

4. Complexity

Practice – Strings

→ Write a recursive method that determines if a string is a palindrome.

Examples:

- * eye
- * mom
- * radar
- * racecar
- * Rise to vote, sir!
- * Never odd or even!
- * A nut for a jar of tuna.
- * Campus Motto: Bottoms up, Mac.
- * Ed, I saw Harpo Marx ram Oprah W aside!
- * Doc note: I dissent. A fast never prevents a fatness. I diet on cod.

Assumptions: non-null input string, all spaces and punctuation removed, all lower-case

Useful string methods:

- * char charAt(int index)
- * int length()
- * String substring(int begin, int one_past_last)

Practice – Complexity of Recursive `isPalindrome`

Problem size N is

1. Equations

2. Table

3. Verify

4. Complexity

Towers of Hanoi

Algorithm

```
solveTowers(count, src, dest, spare) {
```

Complexity

Problem size N is

1. Equations

2. Table

3 Verify

4. Complexity