

CS 240

Data Structures and Algorithms I

Alex Vondrak

`ajvondrak@csupomona.edu`

October 5, 2011

Java Review

Where We Stand

- **Problem:** Not taught Java
 - **Solution:** The textbook gives a solid review in Chapters 1 and 2.
 - **Solution:** Practice; experiment at a PC (e.g., examples from class)
- **Problem:** Taught Java, but can't remember syntax
 - **Solution:** The textbook gives a solid review in Chapters 1 and 2.
 - **Solution:** Practice; experiment at a PC (e.g., examples from class)
- **Problem:** Not taught about Scanners
 - **Solution:** Practice & experiment; maybe on future homework
 - **Solution:** May provide I/O for at least the first project, to ease into it

Algorithms

Definition (Algorithm)

A precise step-by-step plan for a computational procedure that begins with an input value and yields an output value in a finite number of steps

Example (The Searching Problem)

Input: Any array of **ints**, plus a single **int** to search for.

Output: The value **true** if the **int** is an element of the array, or the value **false** if it is not.

```
boolean search(int needle, int [] haystack) {  
    for(int element : haystack)  
        if (element == needle) return true;  
    return false;  
}
```

Comparing Algorithms

- Correctness

Example (The Searching Problem)

```
boolean incorrect(int needle, int[] haystack) {  
    return (haystack[0] == needle);  
}
```

- Speed

Problems

How do we measure speed?

- `System.currentTimeMillis()`
- \$ time ...

Counting Seconds vs Counting Steps

Counting Seconds

Literal running times vary...

- More/less sophisticated hardware (e.g., processor)
- More/less sophisticated software (e.g., OS, compiler, etc.)
- Random chance (e.g., what your OS is doing at the moment)

Counting Steps

To normalize our units of comparison, we can agree to count the total number of “primitive” operations an algorithm performs.

But what is “primitive”?

Counting Steps

A Literal Example

Definition (Eiffel Tower Problem)

- You and a friend are at the top of the Eiffel Tower
- You want to count how many steps there are to the bottom (2689)
- What are the primitive operations?
 - Stepping on a single stair step
 - Marking a single character on a piece of paper

Example (Algorithm 1)

- 1 Take the paper, and go down the stairs
- 2 Every time you take a step, put a tally mark on the paper
- 3 At the bottom, climb back to the top and give your friend the paper

How many primitive operations do you perform?

Counting Steps

A Literal Example

Definition (Eiffel Tower Problem)

- You and a friend are at the top of the Eiffel Tower
- You want to count how many steps there are to the bottom (2689)
- What are the primitive operations?
 - Stepping on a single stair step
 - Marking a single character on a piece of paper

Example (Algorithm 2)

- 1 Take one step down, place your hat upon it
- 2 Go back to the top and tell your friend to mark a tally
- 3 Go down to your hat, take one more step, and repeat the process

How many primitive operations do you perform?

Counting Steps

A Literal Example

Definition (Eiffel Tower Problem)

- You and a friend are at the top of the Eiffel Tower
- You want to count how many steps there are to the bottom (2689)
- What are the primitive operations?
 - Stepping on a single stair step
 - Marking a single character on a piece of paper

Example (Algorithm 3)

- 1 You see another friend at the bottom of the staircase
- 2 He shows you a sign with the number of steps in decimal
- 3 You write down each digit on the piece of paper

How many primitive operations do you perform?

Counting Steps

In Code

Primitive operations on most modern processors include:

- Arithmetic (e.g., +, -, *, /)
- Conditionals (e.g., **if**, ==)
- Fetching/storing a single location in memory (e.g., setting a variable)

Example (The Searching Problem)

- Let $t_i = \#$ times **for** gets executed at element i .
- Let $n = \text{haystack.length}$.

	Cost	Times
<code>for(int element : haystack)</code>	C_1	$\sum_{i=0}^{n-1} t_i$
<code>if (element == needle)</code>	C_2	$\sum_{i=0}^{n-1} t_i$
<code>return true;</code>	C_3	1 or 0
<code>return false;</code>	C_4	1 or 0