

## CSC207.01 2013F, Class 30: An Introduction to Sorting

---

### Overview

- Preliminaries.
  - Exam 1 returned. Makeups due Thursday at 10:30 p.m.
  - Admin.
  - Questions on HW7
  - A few notes on Exam 1
- The problem of sorting.
- An object-oriented approach.
- Testing our sorting algorithm.

### Admin

- I will return exam 1 at the beginning of class.
- I returned HW 6 via email. I still plan to grade HW 5. I will not grade HW 2-4, but will give you all checks. (Sorry, it doesn't seem worth grading at this point.)
- Today we will continue the approach of "think for a few minutes, share with the class; if there's time left, repeat". I have some lab activities planned for tomorrow, but also some full class activities.
- Upcoming extra credit opportunities:
  - Study abroad in Budapest session, Wednesday, Lunch, Science 3821
  - Learning from Alumni, Thursday: Jordan Shkolnick '11 (Microsoft)
  - CS Extras, Thursday: ??? (Maybe not)
  - CS Table, Friday: Forthcoming

### HW 7

How do we deal with predicates in search/select

From the exam ...

```
public interface Predicate<T> {
    public boolean test(T val);
}
```

So, for testing ...

```
public class IsEven implements Predicate<Integer> {
    public boolean test(Integer val) {
        return (val % 2) == 0;
    }
}
```

Can we make doubly- circularly-linked lists? Yes.

*Notes on Exam 1*

- Unit testing and the DNF
- Analyzing run times
  - $1 + 2 + 3 + 4 + \dots = n(n+1)/2$ . Remember Gauss
  - $1 + 2 + 4 + 8 + \dots + 2^k = 2^{k+1} - 1$ . Thanks What credit do we get for redoing problems?
  - The average of the old and new grade

*Was Sam overly harsh on HW6?*

- No, but he'll change most of the grades. If you wrote correct code, check, even if you didn't do two versions.

## The problem of sorting

- We have a bunch of stuff and want to put it in order.
- Questions:
  - What type is "bunch of stuff"? Something ordered, like an array or list
  - What does "in order" mean? It depends on the type. For strings it could be alphabetical, or size, or ...
    - In Java, we could use the "natural" order, using compareTo
      - Identifiable because our code crashes when compareTo doesn't exist
      - Or Java won't compile
      - Or Eclipse underlines
      - Or use Comparables
    - Use Comparators

## An object-oriented approach

- Let's say I want to write N different sorting algorithms and be able to substitute in the sorting algorithm of choice. What do I do?
- Design classes, interfaces, and methods

Signature for the sort method?

```
/**
 * Sort an array in place, using comp to arrange values.
 *
 * @pre
 *   We can safely apply comp to any pair of elements in unsorted.
 *   A.k.a "For all, i, j, 0 <= i, j < unsorted.length,
 *   comp(unsorted[i],unsorted[j]) does not throw an exception.
 * @post
```

```
*   For all reasonable i, comp.compare(unsorted[i],unsorted[i+1]) <= 0
*   unsorted is a permutation of the original unsorted
*/
public void sort(T[] unsorted, Comparator<T> comp);
```

How do we 'package' it up (not put it in a Java package)?

## Testing our sorting algorithm

Copyright (c) 2013 Samuel A. Rebelsky.



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.