# CS 211: Project 6 Discussion

Chris Kauffman

Week 13

# Front Matter

## Today

- Discuss P6
- Search/Sort/Comparisons

## Lab 13: Exercises, Recursion

## P6: Crawl / Index

- Due ~ 2 weeks
- Overview of classes, architecture

## End Game

| | | |
|---|---|---|
| 4/24 | Mon | P6, Comparisons |
| 4/26 | Wed | Recursion |
| | | Lab 13 Recursion |
| 5/1 | Mon | Stacks/Queues |
| 5/3 | Wed | Review/Evals |
| | | Lab Review/Evals |
| 5/7 | Sun | P6 Due |
| Mon | 5/15 | Final Exams |
| | 002 | 10:30am-1:15pm |
| | 006 | 1:30pm-4:15pm |

# Key Broad Concepts

## HTML Parsing

Use the `jsoup` library for HTML parsing

```
File input = new File("crawls/small-site/start.html");
Document doc = Jsoup.parse(input, "UTF-8");
ArrayList<Element> links = doc.select("a[href]");
String text = doc.body().text();
```

## Binary Search and Comparable

- Use `binarySearch(..)` methods from the `Arrays` class and `Collections` class
- Should not need to `sort(..)` if you use `ArraySet` and insert in sorted order
- `IndexEntry` will implement `Comparable` to be in an `ArraySet`

# ArraySet

- A set of unique objects: no duplicates
- Stored in sorted order, enables binary search to be used
- Stores `Comparable` stuff

```
> ArraySet<String> bs =
   new ArraySet<String>();
> bs.add("R")
true
> bs.add("R") // duplicate
false
> bs
[R]
> bs.add("C");
> bs.add("G");
> bs.add("A");
> bs.add("X");
```

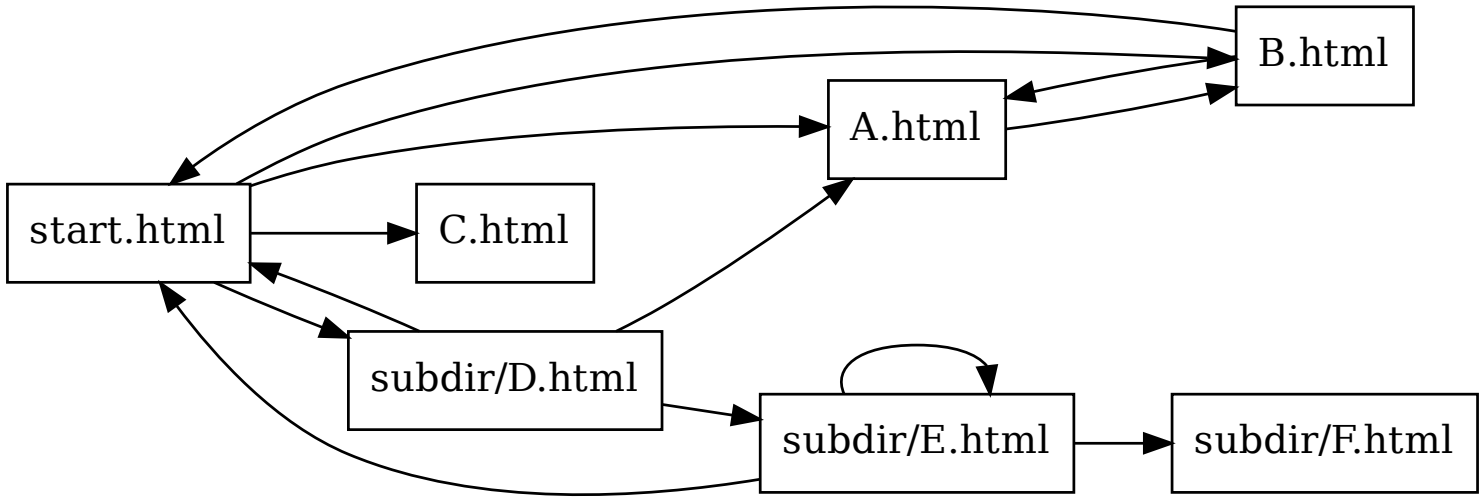
```
> bs
[A, C, G, R, X]
> bs.get("X")
X
> bs.contains("T")
false
> bs.get("T")
null
```

# Crawlers

- Represents a way to crawl through a set of pages
- Keeps track of a set of `foundPages`
- Classes form a small hierarchy

```
Crawler : abstract parent which leaves the crawl( ) method to children
|
+--RecursiveCrawler : crawl through links using recursion
|
+--IterativeCrawler : crawl through links without recursion
|
+--MockCrawler      : (provided) doesn't actually crawl but
                      can test other Crawler methods
```

# Example Crawl




- Starting at `start.html`
- Demonstrate recursive crawl
- Demonstrate iterative crawl

# Page Index

- Goal of any crawl: create data that gives words on pages
- Each `IndexEntry` is

```
@ bread
crawls/small-site/A.html
crawls/small-site/B.html
crawls/small-site/C.html
```

- `PageIndex` is a set of `IndexEntries`
- Doing

```
index.query("bread acrobat")
```

results in

```
crawls/small-site/A.html
```

```
INDEX: 37 entries
--------------------
@ acrobat
crawls/small-site/A.html
crawls/small-site/subdir/E.html
crawls/small-site/subdir/F.html
@ alert
crawls/small-site/B.html
@ argyle
crawls/small-site/A.html
@ bored
crawls/small-site/B.html
@ bread
crawls/small-site/A.html
crawls/small-site/B.html
crawls/small-site/C.html
@ champagne
crawls/small-site/C.html
crawls/small-site/subdir/D.html
...
```