CSC148 winter 2014
sorting, recursion limits
week 11

Danny Heap
heap@cs.toronto.edu
BA4270 (behind elevators)
http://www.cdf.toronto.edu/~heap/148/W14/
416-978-5899

March 24, 2014
Outline

$O(n \lg n)$ sorts compared
You will have a chance in lab to tweak `merge_sort`, `quick_sort`, and `tim-sort` (Python’s built-in sort). You can get some idea of how they scale by running `sort.py`

- why does `tim-sort` do so well?

  think **radix sort**

- what is with `count_sort` anyway?
Some programming languages implement the simplest recursions as loops, but Python doesn’t. One consequence is that our first draft of _contains_ can easily exceed the recursion depth. Rewrite it with while
redundant function calls

The most intuitive version of fibonacci ends up making many redundant function calls:

```python
def fib(n):
    """Return the nth fibonacci number"""
    if n < 2:
        return n
    else:
        return fib(n - 1) + fib(n - 2)
```

\[
f(30) = f(28) + f(29)
\]

\[
f(28) + f(27) + f(27) + f(26)
\]

\[
f(26) + f(27) + f(26) + f(26)
\]

\[
f(26) + f(27)
\]
Never compute the same thing twice (if you can help it)!

memoize!
test 2 coverage...

Everything since test 1:
- linked lists (more than one implementation)
- linked binary trees
- binary search trees
- big-Oh sorting algorithms, scaling with size

labs, lectures, + exercises

solutions to recent lab