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

$\mathcal{O}(n \lg n)$ 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?

- what is with **count_sort** anyway?

running out of stack

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:

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

memoize!

Never compute the same thing twice (if you can help it)!

test 2 coverage. . .

Everything since test 1:

- linked lists (more than one implementation)
- linked binary trees
- binary search trees
- big-Oh