bash today, C tomorrow

- Quick reprise: debugging, performance
- What's homework 2B? (yes, it's posted)
- Some looks at solutions to 2A

Debugging

- "Debugging is important, especially since the shell is so sensitive to details. I recommend two things: (a) trying your commands individually in the command-line as you're trying to build your shell scripts; and (b) assigning and echoing 'unnecessary' variables in your scripts that can be used to help see what's happening step-by-step."

- When things don't work, what do you do?

Performance

- I'm not worried about performance (within a little bit of reason) on 2A. Bill Wulf, who served as president of the National Academy of Engineering for over a decade, once said something like: "More mistakes are made by premature optimization than for any other reason including sheer ignorance."
  - OK, maybe it doesn't work right, but at least it's really fast.
  - Well, if it doesn't have to work right, I can make it even faster!

Algorithmic complexity

- When dealing with a lot of data, what is usually most important about performance is the underlying algorithmic complexity
  - Very roughly, how many times do you need to touch each data item
  - Examples
    - Finding a number in an unsorted list: linear search
    - Finding a number in a sorted list: linear or binary search
    - Sorting a list: \(O(N^2)\) vs. \(O(N \log N)\)
  - HW2: if you touch every entry in the dictionary many times for each input string, that might be a problem – there are 479,829 entries

Performance: one more thing

- Once you get the algorithmic complexity "right", there can still be many ways to improve performance
- A classic example is that some arithmetic operations are faster to execute than others but are equivalent
  - \(x \times 2\) vs. \(x+x\)
  - \(x\) vs. \(\text{left-shift } x\)
  - 001101010 \(\Rightarrow\) 011010100 \([106 \times 2 = 212]\)
- Another classic example is that some operations are faster than others to execute – for example, creating ("forking") a new process in Unix is generally more expensive than computing in the same process
- These, however, require some actual knowledge about the costs factors you face – without that or at least significant experience, you're likely to guess wrong about what is costly
Questions?