Chapter 8, Strings:

- General introduction; string sorting (**Today**)
- Tries (Friday)
- Other string topics (next week Monday and Wednesday)
  - Regular expressions
  - ~~Huffman encoding~~
  - ~~Edit distance~~
  - ~~Grammars and parsing~~

Today:

- End-of-semester business
- Sorting strings
  - Why we care about strings
  - String quick sort
  - String bucket sort
  - String radix sort

Projects:

- Last regular project score update on Tues, Dec 6
- "Two minute warning" run of scripts on Fri, Dec 9 (*no Schoology update—see report file*)
- All projects due on the last day of *classes*, **midnight between Fri, Dec 9 and Sat, Dec 10**—*not last day of finals*.

Final exam

- Our final exam block is **Wed, Dec 14, 10:30am–12:30pm**
- During our final exam block, we will meet in the **CSCI lab**
- Test 3 ("written"/conceptual part) will be like Test 1, but covering BSTs (ch 5) through strings (ch 8)
- Test 4 (programming part) will work the same way as Test 2, covering dynamic programming, hashing, and strings.

Why we care about strings

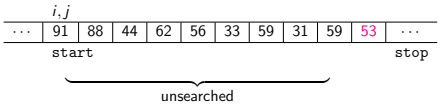
- Strings are different
- Strings are common
- Strings are a representative example

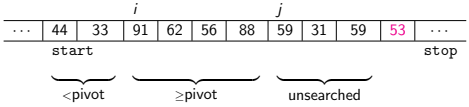
```
public class DNASequence {
    /** An alphabet for DNA */
    private static enum Nucleotide { A, C, G, T }
    /** The string of nucleotides */
    private Nucleotide[] sequence;
}
```

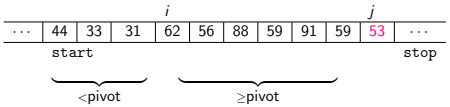
```
public class BigInt {

    private byte[] digits;

    /** Compute the sum of this and another BigInt. */
    public BigInt add(BigInt other) {
        // The result object
        BigInt sum = new BigInt();
        // The result object has at most one more digit
        // than the larger number of digits of the two addends
        sum.digits = new byte[(digits.length > other.digits.length?
                digits.length : other.digits.length) + 1];
        // Add by column
        int carry = 0;
        for (int i = 0; i < sum.digits.length; i++) {
            // Digits in current columns of the two addends
            int a = digits.length <= i? digits[i] : 0;
            int b = other.digits.length <= i ? other.digits.length : 0;
            // The sum of the current digits plus carry from previous iteration
            int s = a + b + carry;
            // Mod that sum by 256 to get the appropriate digit in result,
            // divide to get the carry for next time.
            sum.digits[i] = (byte) (s % 256);
            carry = s / 256;
        }
        assert carry == 0;
        return sum;
    }
}
```

```
struct employee
{
   char surname[20];
   char first_name[20];
   double salary;
   char extension[4]
};

struct book
{
    char title[100];
    char author[50];
    int pages;
    char call_number[8];
    int status;
};

struct complex_number { double real, double imag };
```

Quick sort:







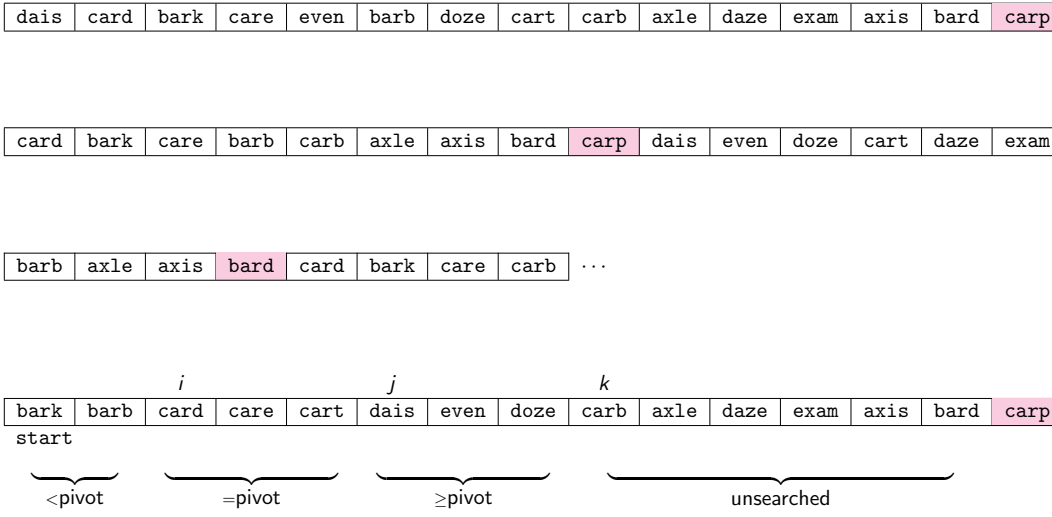
Invariant 11 (Loop of `partition()`)

(a) $\texttt{start} \leq i \leq j < \texttt{stop}$.

(b) $\forall\ k \in [\texttt{start}, i)$, $\texttt{sequence}[k] < \texttt{sequence}[\texttt{stop} - 1]$.

(c) $\forall\ k \in [i, j)$, $\texttt{sequence}[k] \geq \texttt{sequence}[\texttt{stop} - 1]$.

(d) $j - \texttt{start}$ is the number of iterations completed.


dais card bark care even barb doze cart carb axle daze exam axis bard carp
card bark care barb carb axle axis bard carp dais even doze cart daze exam
barb axle axis bard card bark care carb ···
i
j
k
bark barb card care cart dais even doze carb axle daze exam axis bard carp
start
stop
<pivot
=pivot
≥pivot
unsearched

Invariant 37. [Loop of `string_quick_sort_r()`]

Let c be the character in position `pre` in the string in position `stop` $- 1$.

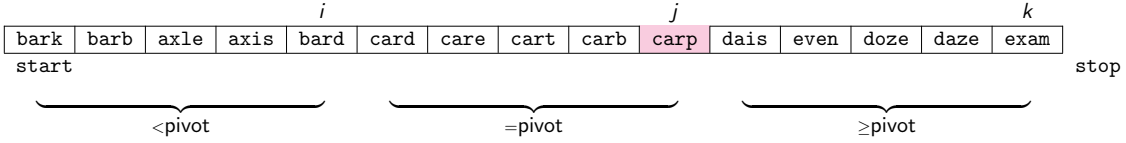
(a) `start` $\leq i \leq j \leq k <$ `stop`

(b) (Informal) For all the strings in range [`start`, i), their character in position `pre` is less than c.

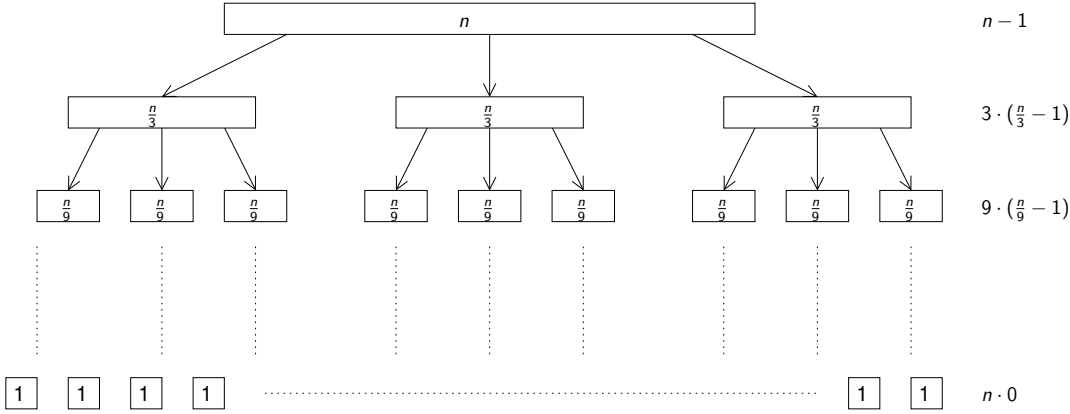
(c) (Informal) For all the strings in range $[i, j)$, their character in position `pre` is equal to c.

(d) (Informal) For all the strings in range $[i, j)$, their character in position `pre` is greater than to c.

(e) $k -$ `start` is the number of iterations completed.



Invariant 38. [Precondition of `string_quick_sort_r()`]
$\forall\ i, j \in [\texttt{start}, \texttt{stop}), \forall k \in [0, \texttt{pre}), \texttt{sequence}[i][k] = \texttt{sequence}[j][k]$.



dais	card	bark	care	even	barb	doze	cart	carb	axle	daze	exam	axis	bard	carp

barb	carb	card	bard	care	doze	axle	daze	bark	exam	even	carp	dais	axis	cart

exam	even	dais	axis	axle	barb	carb	card	bard	care	bark	carp	cart	doze	daze

dais	barb	carb	card	bard	care	bark	carp	cart	daze	doze	even	exam	axis	axle

axis	axle	barb	bard	bark	carb	card	care	carp	cart	dais	daze	doze	even	exam

beach	event	can	core	hope	any	front	ball	done	a	frond	an	i	give	eve

can	core	hope	any	ball	done	a	an	i	give	eve	frond	beach	event	front

can	any	a	an	i	eve	beach	core	hope	done	give	ball	frond	event	front

a	an	i	beach	eve	event	ball	can	done	frond	front	hope	core	give	any

a	i	ball	can	beach	give	an	any	done	hope	core	frond	front	eve	event

a	an	any	ball	beach	can	core	done	eve	event	frond	front	give	hope	i

Coming up:
Do **Open Addressing** *project (suggested by Friday, Dec 2)*

Due **Wed, Nov 30** *(end of day)*
Read Section 8.1
Do Exercises 8.(4 & 5)

Due **Thurs, Dec 1**
Take quiz (on Section 8.1)

Due **Fri, Dec 2**
Do Exercises 8.(7, 14, 20)
Read Section 8.2