

## Chapter 8, Strings:

- ▶ General introduction; string sorting (**Today**)
- ▶ Tries (next wee Monday)
- ▶ Other string topics (next week 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, Apr 25
- ▶ “Two minute warning” run of scripts on Fri, Apr 28 (*no Schoology update—see report file /cslab/class/cs345/(your userid)/(your userid).report*)
- ▶ All projects due on the last day of *classes*, **midnight between Fri, Apr 28 and Sat, Apr 29**—*not last day of finals.*

## Final exam

- ▶ Our final exam block is **Wed, May 3, 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 the second part (MST and SSSP) of graphs (ch 6), and then 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;  
}
```

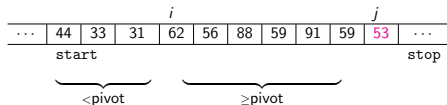
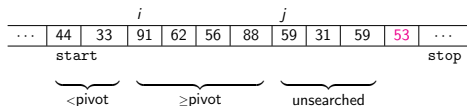
```
public class BigInt {  
  
    private byte[] digits;  
  
    /** Compute the sum of this and another BigInt. */  
    public BigInt add(BigInt other) {  
        // The result object  
        BigInt sum = new BigInt();  
        // The result object has at most one more digit  
        // than 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[i] : 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:



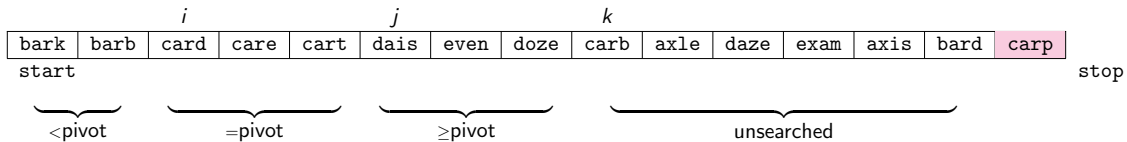
### Invariant 11 (Loop of partition())

- (a)  $\text{start} \leq i \leq j < \text{stop}$ .
- (b)  $\forall k \in [\text{start}, i)$ ,  $\text{sequence}[k] < \text{sequence}[\text{stop} - 1]$ .
- (c)  $\forall k \in [i, j)$ ,  $\text{sequence}[k] \geq \text{sequence}[\text{stop} - 1]$ .
- (d)  $j - \text{start}$  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
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

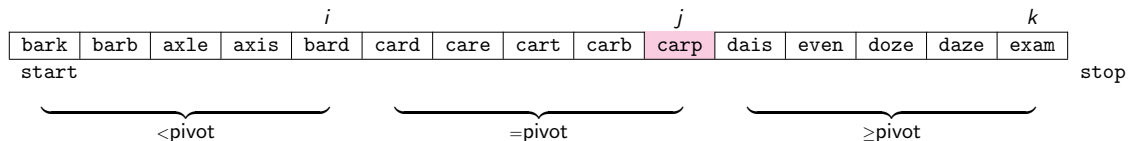
barb	axle	axis	bard	card	bark	care	carb	...
------	------	------	------	------	------	------	------	-----



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

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

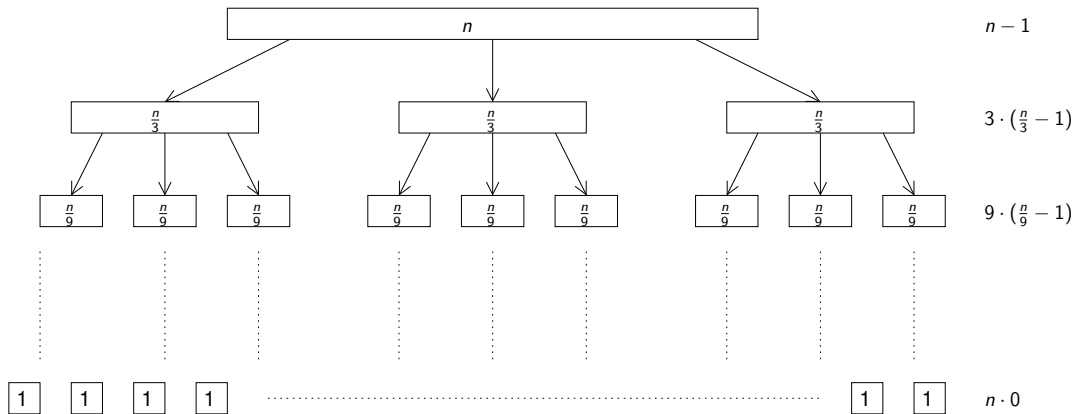
- (a)  $\text{start} \leq i \leq j \leq k < \text{stop}$
- (b) (Informal) For all the strings in range  $[\text{start}, i)$ , their character in position `pre` is less than  $c$ .
- (c) (Informal) For all the strings in range  $[i, j)$ , their character in position `pre` is equal to  $c$ .
- (d) (Informal) For all the strings in range  $[j, k)$ , their character in position `pre` is greater than  $c$ .
- (e)  $k - \text{start}$  is the number of iterations completed.





### Invariant 38. [Precondition of `string_quick_sort_r()`]

$\forall i, j \in [\text{start}, \text{stop}), \forall k \in [0, \text{pre}), \text{sequence}[i][k] = \text{sequence}[j][k].$



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 **Perfect Hashing** project (*suggested by Wednesday, Apr 26*)

Due **Fri, Apr 21** (*end of day*)

Read Section 8.1

Do Exercises 8.(4 & 5)

Take *the last* quiz

(*Accompanying programming practice problem due next week Monday*)

Due **Mon, Apr 25** (*end of day*)

Do exercises 8.(7, 14, & 20) (*from Section 8.1*)

Read Section 8.2

(*No quiz*)