Chapter 3, Case Studies:

- Linear-time sorting algorithms (**Today** and Wednesday)
- Disjoint sets and array forests (Friday)
- Priority queues (Next week Monday)
- N-sets and bit vectors (Next week Wednesday)

Today (and Wednesday):

- Recent quiz problem
- Iterators in adapter data structures
- Intro to "case studies"
- Limitations of comparison-based sorting
- Counting sort
- Radix sort

	ArrayList	LinkedList
`get()`		
`set()`		
`add()`		

Can't you tell a good tree from a poor tree?

I have just been thinking, and I have come to a very important decision. These are the wrong sort of bees.

Good sorts	**Bad sorts**
Merge	Selection
Quick (expected case)	Insertion
Shell (unassigned project)	Bubble
Heap (Section 3.3)	

The ultimate measure for algorithms is their running time, but counting the number of comparisons between elements in the sequence is a reasonable proxy for running time when comparing sorting algorithms. At least we can say that if the expected case of a sorting algorithm makes $\Theta(n \lg n)$ comparisons, then its running time must be $\Omega(n \lg n)$—that is, it can't be better than $n \lg n$.

It turns out that $\Theta(n \lg n)$ is in fact the best we can do for the expected case of sorting algorithms that make decisions by comparing elements in the sequence. Put formally,

> **Theorem 1.** *If T is an algorithm that sorts a sequence by comparing pairs of elements in the sequence, then the expected running time of T is $\Omega(n \lg n)$.*

1 0 1 1 4 0 2 1 3 0 1 1 3 2 2 1 2 1 4 0 4 2 3 1 1 2 1 1 2 1 3 2 4 0 4

0. Alice 0
1. Bob 2
2. Carol 4
3. Dave 4
4. Eve 2
5. Fred 0
6. Georgia 0
7. Henry 1
8. Ida 4
9. Jack 2
10. Karen 4
11. Larry 0
12. Moira 2
13. Nate 3
14. Olivia 1
15. Pete 1
16. Queenie 1
17. Ralph 4
18. Sara 2
19. Trent 4
20. Ursulla 2
21. Vick 3
22. Wendy 1
23. Xavier 2
24. Yvette 0
25. Zeke 3

Coming up:

Do "basic data structures" practice problems (suggested by today, Mon, Jan 31)
Do **"implementing ADTs"** *project (suggested by Wed, Feb 2)*

Due **Wed, Feb 2***: (class time)*
Read Section 3.1
Do Exercises 2.(21–23)
Take sorting quiz

Due **Fri, Feb 4***: (end of day)*
Read Section 3.2
Do Exercises 2.(12 & 15) and 3.8.
Take disjoint sets quiz

Invariant (Loop of `radix_sort`)

(a) i *is the number of iterations completed.*

(b) $\texttt{r\_pow} = \texttt{r}^{\texttt{i}}$.

(c) $\forall\ k \in [0, n-1), \texttt{sequence}[k] \mod \texttt{r}^{\texttt{i}} \leq \texttt{sequence}[k+1] \mod \texttt{r}^{\texttt{i}}$

0110	1011	1100	0111	0001	1110	1001	1101

0110	1011	1100	0111	0001	1110	1001	1101

0110	1100	1110	1011	0111	0001	1001	1101

0110	1011	1100	0111	0001	1110	1001	1101

0110	1100	1110	1011	0111	0001	1001	1101

1100	0001	1001	1101	0110	1110	1011	0111

0110	1011	1100	0111	0001	1110	1001	1101

0110	1100	1110	1011	0111	0001	1001	1101

1100	0001	1001	1101	0110	1110	1011	0111

0001	1001	1011	1100	1101	0110	1110	0111

0110	1011	1100	0111	0001	1110	1001	1101

0110	1100	1110	1011	0111	0001	1001	1101

1100	0001	1001	1101	0110	1110	1011	0111

0001	1001	1011	1100	1101	0110	1110	0111

0001	0110	0111	1001	1011	1100	1101	1110