Chapter 3, Case Studies:

- Linear-time sorting algorithms (last week Monday and Wednesday)
- Disjoint sets and array forests (last week Friday)
- Priority queues (Monday and Today)
- $N$-sets and bit vectors (Today)
- (Start graphs Friday)

Today:

- Problem statement
- Abstractions and insights
- Project tips
List

Set

<<interface>>

add(E)
contains(E)
remove(E)
size()
isEmpty()

LinkedList

BArrayNSet

BitVecNSet

NaiveNSet

NSet

<<interface>>

range()
complement()
union()
intersection()
difference()
\{1, 3, 4, 11\} \subseteq [0, 16)
<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitwise AND</td>
<td>result bit is set if both operand bits are set</td>
<td>0100110 &amp; 1101011 = 0100010</td>
</tr>
<tr>
<td>Bitwise OR</td>
<td>result bit is set if at least one operand bit is set</td>
<td>0100110</td>
</tr>
<tr>
<td>Bitwise XOR</td>
<td>result bit is set if exactly one operand bit is set</td>
<td>0100110 ^ 1101011 = 1001101</td>
</tr>
<tr>
<td>Bitwise NEG</td>
<td>flip each bit of the operand</td>
<td>0100110 ~ = 0010100</td>
</tr>
</tbody>
</table>
Coming up: (all end-of-day)

Do linear sorting project (suggested by this past Monday)
Do heaps and priority queue project (suggested by Mon, Feb 14)
Do bit vector and N-set project (suggested by Wed, Feb 23)

Due Today:
Take heap/pq quiz

Due Thurs, Feb 10:
Read Section 3.4
Do Exercises 3.(27 & 28)
Take N-sets quiz

Due Wed, Feb 23 (but spread it out):
Read Section 4.(1–3)
Do Exercises 4.(22-25).
Take graph quiz