Chapter 4, Graphs:
- Concepts and implementation (**Today**)
- Traversal (next week Monday *and in lab Thursday*)
- Minimum spanning trees (next week Wednesday and Friday)
- Single-source shortest paths (Feb 21 and 23)

Today:
- Recent quiz questions
- Applications of graphs
- Vocabulary, taxonomy, and theory
- Representing and implementing graphs
Indicate the worst case running time for each operation in each implementation of a priority queue.

<table>
<thead>
<tr>
<th></th>
<th>ListPriorityQueue</th>
<th>SortedListPriorityQueue</th>
<th>HeapPriorityQueue</th>
</tr>
</thead>
<tbody>
<tr>
<td>insert()</td>
<td>Θ(1)</td>
<td>Θ(n)</td>
<td>Θ(lg n)</td>
</tr>
<tr>
<td>max()</td>
<td>Θ(n)</td>
<td>Θ(1)</td>
<td>Θ(1)</td>
</tr>
<tr>
<td>extractMax()</td>
<td>Θ(n)</td>
<td>Θ(1)</td>
<td>Θ(lg n)</td>
</tr>
<tr>
<td>contains()</td>
<td>Θ(n)</td>
<td>Θ(n)</td>
<td>Θ(n)</td>
</tr>
</tbody>
</table>
3.26 In the NaiveNSet, why does the add() method have an @Override annotation but range(), complement(), union(), intersection(), and difference() do not?

3.27 Explain the + 1 in the array creation new byte[range / 8 + 1] in the BitVecNSet constructor.
Graph
- Vertex (compare *node*)
- Edge (compare *link*)
- Incident
- Adjacent
- Degree
- Complete
- Dense

Sparse
- Directed graph
- Undirected graph
- Parallel edge
- Self loop
- Simple graph
- Weighted graph
Adjectives

- **Trivial**: Having only one vertex and no edges.
- **Simple**: Having no repeated *vertices* (except, possibly, the initial and terminal).
- **Closed**: Having the same vertex as initial and terminal.

Nouns

- **Walk**: An alternating sequence of vertices and edges, each edge coming between its end points.
- **Path**: A walk with no repeated *edge* (repeated vertices are ok).
- **Circuit**: A closed path (no repeated edges, initial and terminal the same).
- **Cycle**: A simple circuit (no repeated edges or vertices, except the initial and terminal, which are the same).
<table>
<thead>
<tr>
<th></th>
<th>Adjacency matrix</th>
<th>Adjacency list</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Space</strong></td>
<td>$\Theta(V^2)$</td>
<td>$\Theta(V + E)$</td>
</tr>
<tr>
<td><code>adjacent(u, v)</code></td>
<td>$\Theta(1)$</td>
<td>$\Theta(deg(u))$ (expected case)</td>
</tr>
<tr>
<td><code>getAdjacents(u)</code></td>
<td>$\Theta(V)$</td>
<td>$\Theta(deg(u))$</td>
</tr>
</tbody>
</table>
Coming up:

*Do heaps and priority queue project (suggested by Mon, Feb 13)*
*Do bit vector and N-set project (suggested by Wed, Feb 15)*

Due **Wed, Feb 15** (but spread it out):

*Read Section 4.(1–3)*
*Do Exercises 4.(26-29).*
*Take graph quiz*