Chapter 4, Graphs:

- Concepts and implementation (Today)
- Traversal (Wednesday)
- Minimum spanning trees (Friday and next week Monday)
- Single-source shortest paths (next week)

Today:

- Applications of graphs
- Vocabulary, taxonomy, and theory
- Representing and implementing graphs
- 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

<table>
<thead>
<tr>
<th>Adjective</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trivial</td>
<td>Having only one vertex and no edges.</td>
</tr>
<tr>
<td>Simple</td>
<td>Having no repeated <em>vertices</em> (except, possibly, the initial and terminal).</td>
</tr>
<tr>
<td>Closed</td>
<td>Having the same vertex as initial and terminal.</td>
</tr>
</tbody>
</table>

### Nouns

<table>
<thead>
<tr>
<th>Noun</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk</td>
<td>An alternating sequence of vertices and edges, each edge coming between its end points.</td>
</tr>
<tr>
<td>Path</td>
<td>A walk with no repeated <em>edge</em> (repeated vertices are ok).</td>
</tr>
<tr>
<td>Circuit</td>
<td>A closed path (no repeated edges, initial and terminal the same).</td>
</tr>
<tr>
<td>Cycle</td>
<td>A simple circuit (no repeated edges or vertices, except the initial and terminal, which are the same).</td>
</tr>
</tbody>
</table>
Graph

<<interface>>

numVertices() : int
numEdges() : int
adjacents(v:int) : Iterable<Integer>
adjacent(u:int, v:int)

ALGBuilder
connect(u:int, v:int)
getGraph(): AdjMatrixGraph

AdjListGraph
adjSets: Set<Integer>[
numEdges: int

AdjMatrixGraph
edges: boolean[][]
numEdges: int

AMGBuilder
connect(u:int, v:int)
getGraph(): AdjMatrixGraph
<table>
<thead>
<tr>
<th></th>
<th>Adjacency matrix</th>
<th>Adjacency list</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>$\Theta(V^2)$</td>
<td>$\Theta(V + E)$</td>
</tr>
<tr>
<td>adjacent(u, v)</td>
<td>$\Theta(1)$</td>
<td>$\Theta(deg(u))$ (expected case)</td>
</tr>
<tr>
<td>adjacents(u)</td>
<td>$\Theta(V)$</td>
<td>$\Theta(deg(u))$</td>
</tr>
</tbody>
</table>
Coming up:

- Do heaps and priority queue project (suggested by Wed, Sept 28)
- Do bit vector and N-set project (suggested by Fri, Sept 30)

Due Fri, Sept 30 (but spread it out):
- Read Section 4.(1–3)
- Do Exercises 4.(26-29).
- Take graph quiz