Announcements

HW 0: average ~28
- HW due Friday
(you should have already started!)
Representing Graphs

How big can \( m \) be (in terms of \( n \))?
Anyone we find it or can't search. Search out from there until?

How?

Is there a path from

s to t connected. Given a network, are

Algorithms on graphs
Breadth First Search (BFS)

- L0
- L1 - at distance 2 from S
- L2 - at distance 3 from S
- L3 - at distance 4 from S

Who is connected to S?
Exercise: What does the BFS tree look like for this graph?
If in a BFS tree, can follow the path from a vertex in a level to a vertex in the BFS tree. Why? If there is a level then + in the BFS tree. Use induction on a vertex in Lij. A path from a vertex in Li goes through if it is in the BFS tree. There is a path from it to it + any it. Property: For each v in the BFS tree.
Pseudo-code: (From text)

1. Initialize empty list $L$. Set $\text{counter} \to 0$.
2. While $L \neq \emptyset$ do:
   a. $L \leftarrow \text{BFS}(G, \{v\})$ (Get next vertex).
   b. $L \leftarrow \text{BFS}(G, \{v\})$ (Get next vertex).
3. For each vertex $v \in V$ do:
   a. $\text{DS} \leftarrow \{\emptyset\}$.
   b. For each edge $e = (u, v) \in E$ do:
      i. If $\text{DS} e \notin \text{DS} \Rightarrow \text{DS} \leftarrow \text{DS} \cup e$.
      j. Add vertex $v$ to $\text{DS}$.
4. For each vertex $v \in V$ do:
   a. If $\text{DS} v = \text{false}$ then:
      i. Mark vertex $v$ as discovered.
   b. Add vertex $v$ to $\text{DS}$.

Note: The pseudo-code is incomplete and lacks necessary details for full understanding.
For each vertex, list all adjacent edges.

\[ \text{deg}(v) = \Omega(n) \]

Case 1: \( \geq 3 \) edges

Next, for loops - why?
Let \( L \) be a linear functional. Then for any \( \lambda \) in \( L^* \), there is a corresponding \( x \) in \( X \) such that \( \lambda(x) = \langle x, \lambda \rangle \). Thus, \( L \) can be identified with \( X \). This identification is unique up to a constant factor. Hence, \( L \) is isomorphic to \( X \).