

Graphs

1. Administrative
 - a. Last required lab assignment this week (optional assignment goes out next week)
 - b. Exam available 4pm Wednesday, due 4pm Friday
 - i. Sign in/out from Lorraine Robinson in TCL 303
 - ii. -1 point per minute after 4pm Friday!
 - iii. Review homeworks, odd-numbered problems while studying
 - iv. Review session tonight, 8pm
 - c. CS Information Session tonight 9pm in the common room (with pizza)
 - d. No class Friday 4/25 and 5/2
 - e. Read chapter 16 on graphs
2. Graphs
3. Examples
4. Notation
 - a. Set notation
 - b. Labels
 - c. Degree
 - d. Source
 - e. Sink
 - f. Adjacent vertices
 - g. Adjacent edges
 - h. Cycle
 - i. Acyclic graph
 - j. Planar graph
 - k. v_0 connected to v_1
 - l. Transitive closure
 - m. Connected component
 - n. Strongly connected component
 - o. Complete graph
 - p. Dual of a graph
5. Adjacency list representation
 - a. Code on back of this page
 - b. (See book for an alternative implementation)
 - c. When is this a good representation?

```
public class ListGraph<VertexLabel, EdgeLabel> {
    private ArrayList<Vertex> vertices = new ArrayList<Vertex>();
    private ArrayList<Edge> edges = new ArrayList<Edge>();

    public class Vertex {
        private VertexLabel label;
        private ArrayList<Edge> outgoingEdges = new ArrayList<Edge>();
        private ArrayList<Edge> incomingEdges = new ArrayList<Edge>();
        private Vertex(VertexLabel L) {
            label = L;
        }
    }

    public class Edge {
        private EdgeLabel label;
        private Vertex start;
        private Vertex finish;
        private Edge(EdgeLabel L, Vertex s, Vertex f) {
            label = L;
            start = s;
            finish = f;
        }
    }

    /** Adds a new disconnected vertex */
    public Vertex addVertex(VertexLabel label) {

    }

    /** Adds a directed edge */
    public Edge addEdge(EdgeLabel label, Vertex v0, Vertex v1) {

    }
}
```