

CS367 Lecture 27

Thursday 31 July 2014

Announcements/Reminders:

Last class:

- Sorting (smarter)

Today:

- Sorting (finish)
 - Radix Sort
 - Stable sorts
 - Sorting in Java
- Graphs
- Week in Review

Radix Sort

Strategy:

Analysis:

Stable Sorts

Madison WI
New York NY
Chicago IL
Detroit MI
Buffalo NY
Milwaukee WI
Peoria IL

Sorted by state:

Chicago IL
Peoria IL
Detroit MI
New York NY
Buffalo NY
Madison WI
Milwaukee WI

Chicago IL
Peoria IL
Detroit MI
Buffalo NY
New York NY
Madison WI
Milwaukee WI

Chicago IL
Peoria IL
Detroit MI
New York NY
Buffalo NY
Milwaukee WI
Madison WI

What's the difference?

Sorting in Java (`java.util`)

`Arrays.sort(array_to_sort):`

```
String[] names = {"Tim", "Alison", "Nate"};
Arrays.sort(names);
System.out.println(Arrays.toString(names));
```

`Collections.sort(List):`

Use Comparators to define your own order:

```
public interface Comparator<T> {
    int compare(T o1, T o2);
}

public class myCmp<myClass> implements Comparator<myClass> {
    public int compare(myClass o1, myClass o2) {
        ...
    }
}
```

Types of Data Structures

(in terms of #predecessors and #successors)

Linear

Hierarchical

Graphical

Graph Terminology

- Nodes
- Edges: Weighted and unweighted
- Directed vs. undirected
- Degree of a node
- Self-edge
- Cyclic vs. acyclic
- Adjacency
- Predecessor, successor
- Path, path length
- Search/Traversal
- Complete graphs
- Directed Acyclic Graphs (DAG)
- Connected: Weakly, Strongly

Implementing Graphs

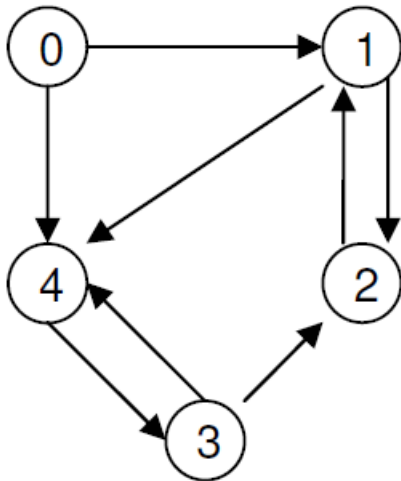
GraphNode

Graph

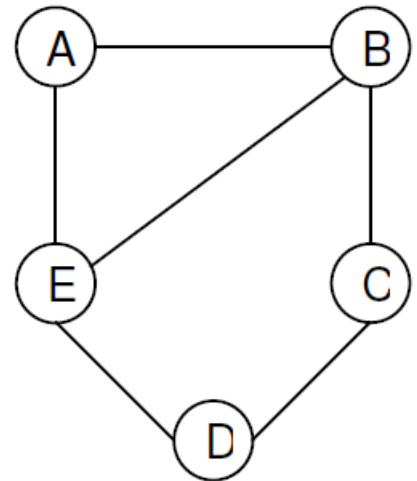
Representing Edges

Adjacency Matrix:

Graph 1



Graph 2



Adjacency matrix for these graphs:

Graph 1

	0	1	2	3	4
0					
1					
2					
3					
4					

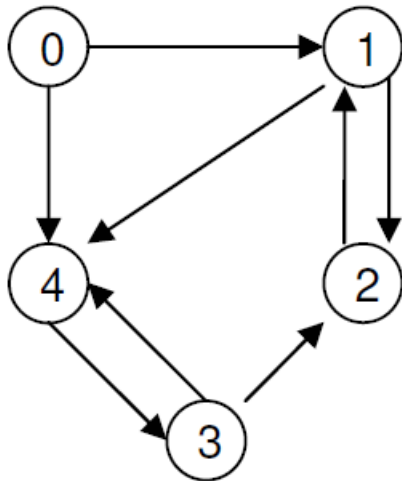
Graph 2

	A	B	C	D	E
A					
B					
C					
D					
E					

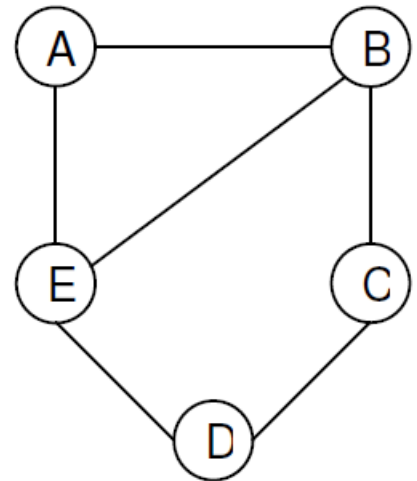
Representing Edges (cont'd)

Adjacency Lists:

Graph 1



Graph 2



Adjacency list for these graphs:

Graph 1

Graph 2

0:		A:	
1:		B:	
2:		C:	
3:		D:	
4:		E:	