Graphs

Representations
Algorithms

Graph representations

1. Adjacency matrix.
2. Adjacency list.
3. Edge Listing.

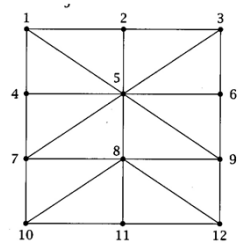
- Write and analyze algorithms for finding the degree of a vertex in representations each representation, in a graph with n vertices and m edges.
  You may assume that the names of the vertices are the numbers 0 through n -1.
- Do the same for finding the degrees of *all* vertices.
- What does the square of an adjacency matrix give us?
- Boolean adjacency matrix?

Traversing (searching) a graph

- Start at one vertex.
- Visit each reachable node only once.
- Until all nodes in the component are visited.
  - Or until we find what we are looking for.
- Search other components if necessary
- Two common approaches:
  - Depth-first (dfs). Extend path as far as possible, then back up.
  - Breadth-first (bfs). visit nodes closest to starting node first.

Depth-first search

- Start at a given node.
- If there is an adjacent unvisted node, visit it.
- If not, back up and see if there are any untried adjacent unvisited nodes.
- Look at an example.

2. 

Start vertex = 3

A ____ -order traversal of a binary tree is a special case of dfs

Algorithm for DFS

- fields of the graph class:
  - BinaryNode [ ] adj // adjacency list
    - length of array is two more than # of vertices (we don't use 0, and do use n to be consistent with our pictures).
    - each BinaryNode contains an index of a vertex.
  - boolean [ ] visited; // initialized to all false.
- ```
  void dfs(int start){
    int n = adj.length-1;
    for (int i=1; i<=n; i++)
       visited[i] = false;
    dfsRecursive(start);
  }
  ```
- **Write dfsRecursive.**

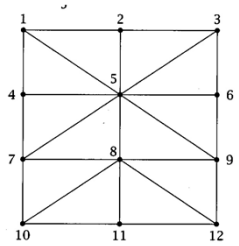
dfsRecursive

```
void dfsRecursive(int start) {
  System.out.println(start); //visit
  visited[start] = true;
  BinaryNode current = adj[start];
  while (current != null) {
    int v = current.element;
    if (! visited[v])
      dfs_recurs(v);
    current = current.next;
  }
}
```

Worst-case running time of dfs?
```

Breadth-first search

- Start at given node with empty queue.
- Add all adjacent unvisited nodes to the queue
- If the queue is nonempty, remove and visit first node, then add all of its adjacent unvisited nodes to end of the list.

2. 

Start vertex = 3

A ____ -order traversal of a binary tree is a special case of bfs.

We will not write the algorithm.
If you are not sure that you could do it, try it!

DFS and BFS animation

- http://www.cs.sunysb.edu/~skiena/combinatorica/animations/search.html