ECE 2574
Introduction to Data Structures and Algorithms

39: Graph Traversals and Algorithms

Chris Wyatt
Electrical and Computer Engineering
Virginia Tech

Traversals and Searching

Traversals and Searching

- Depth-First Search (DFS)
- Breadth-First Search (BFS)
- Best-First Search
- A* Search

} Weighted Graphs

Introduction to Graph Algorithms

Depth-First Traversal

Given an initial vertex V

```
DFS(V)
    mark V as visited
    for each unvisited vertex U adjacent to V
        DFS(U)
```

How can we implement this using recursion?

How can we store the unvisited vertices?

How fast can we mark and test visited?

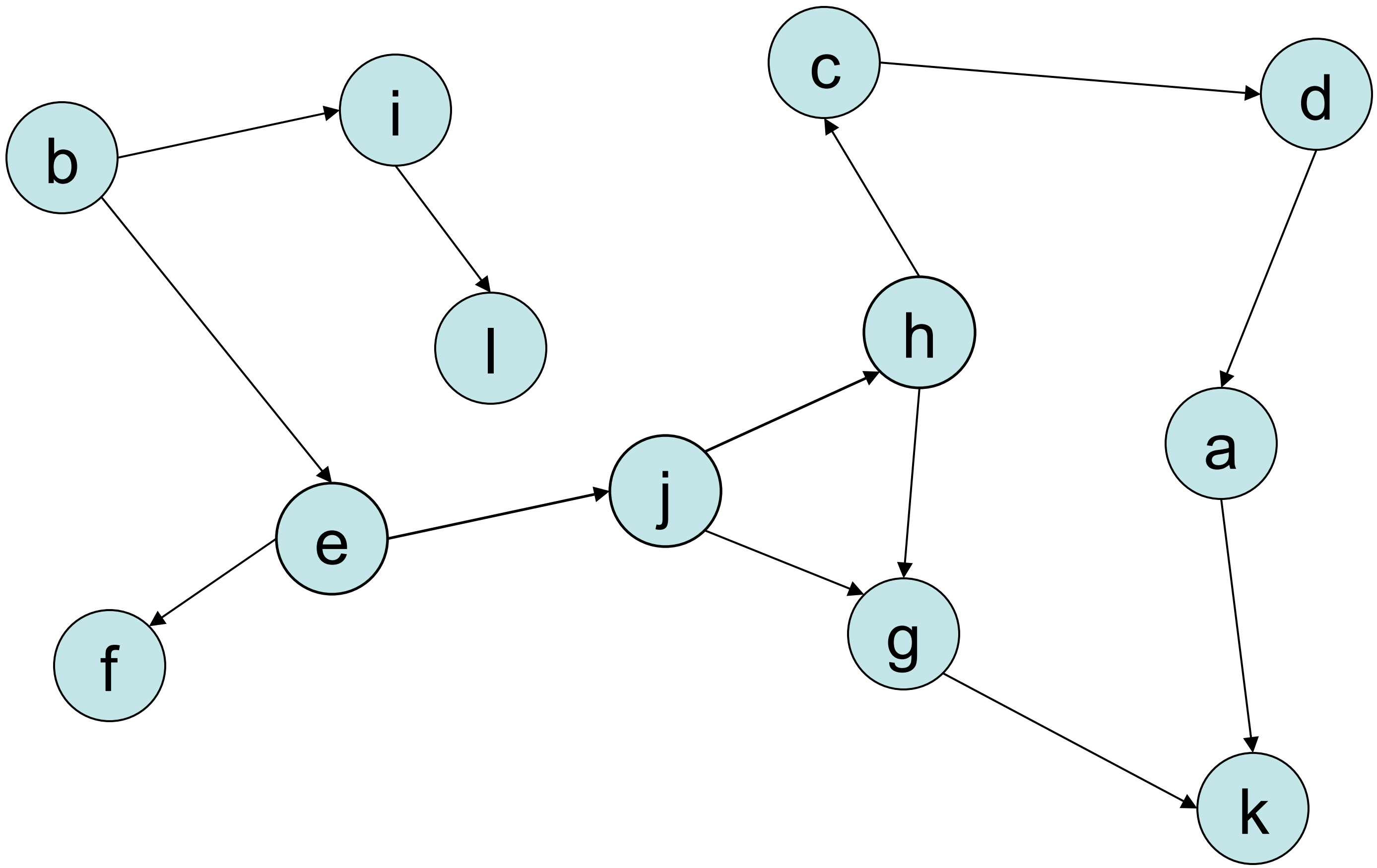
What order should the adjacent vertices be visited?

Stack-based DFS

```
Given an initial vertex V
DFS(V)
    mark V as visited
    for each unvisited vertex U adjacent to V
        push(U)
    while(stack not empty)
        pop -> W
        mark W as visited
        for each unvisited vertex U adjacent to W
            push(U)
```

Depth-First Traversal: example

DFS(b)

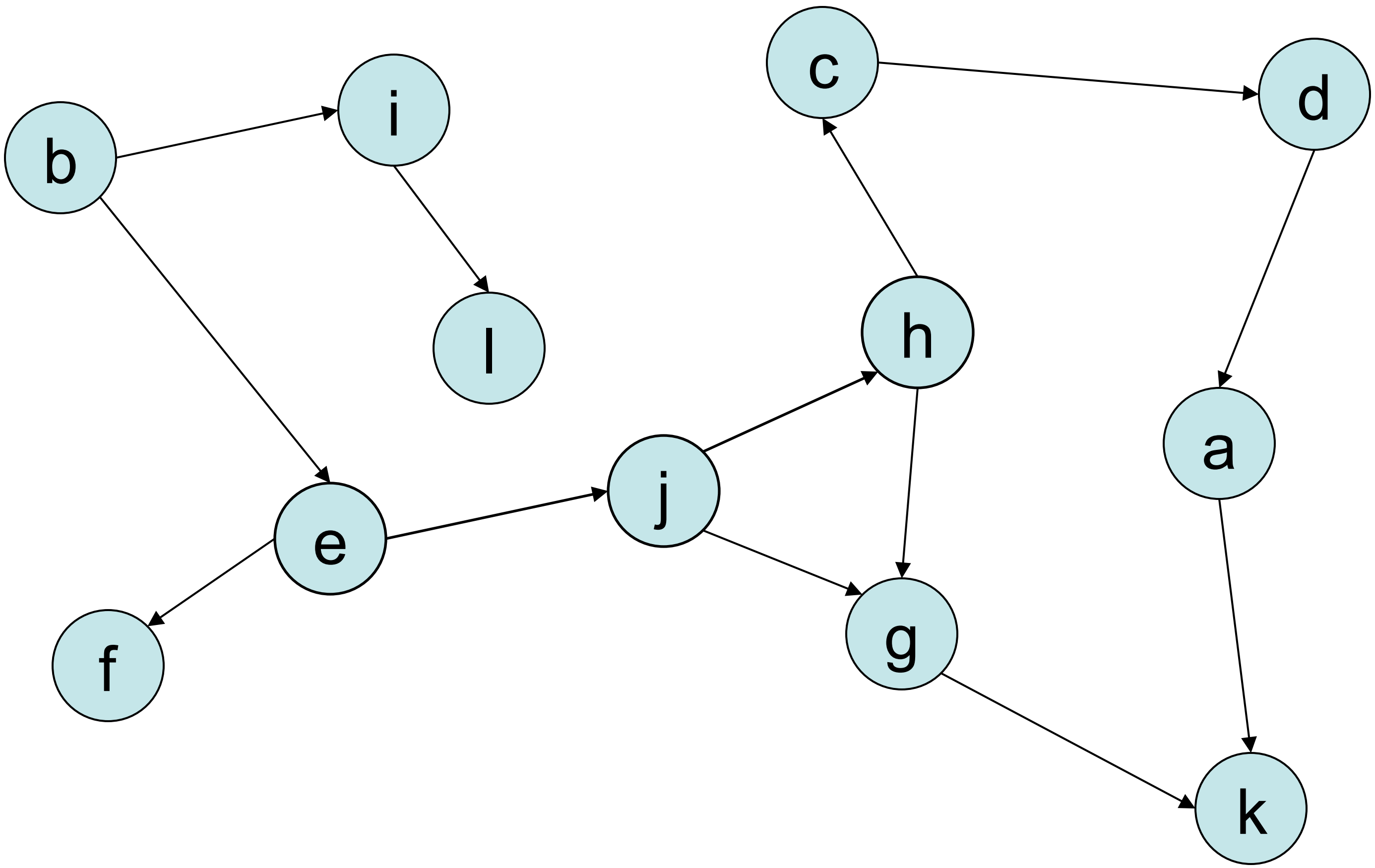


Breadth-First Traversal

```
Given an initial vertex V
BFS(V)
    mark V as visited
    for each unvisited vertex U adjacent to V
        enqueue(U)
    while(queue not empty)
        dequeue -> W
        mark W as visited
        for each unvisited vertex U adjacent to W
            enqueue(U)
```

Breadth-First Traversal: example

BFS(b)



Graph Search Problems

Given a graph rooted at some vertex R with a goal G, searching the graph for G is a common task.

In some cases the path is important

- example: N-puzzle problem

In others it is not

- example: constraint satisfaction problems

Weighted Graphs

In many cases the edges have a cost or weight associated with them (distance for example).

The performance of Graph Search can then be analyzed along the following lines

- Is the solution optimal ?
- Is the solution complete (if the goal exists it is found)?

Best-First Search

DFS and BFS are called *uninformed* because they simply expand nodes (into the stack or queue) in the same or arbitrary order.

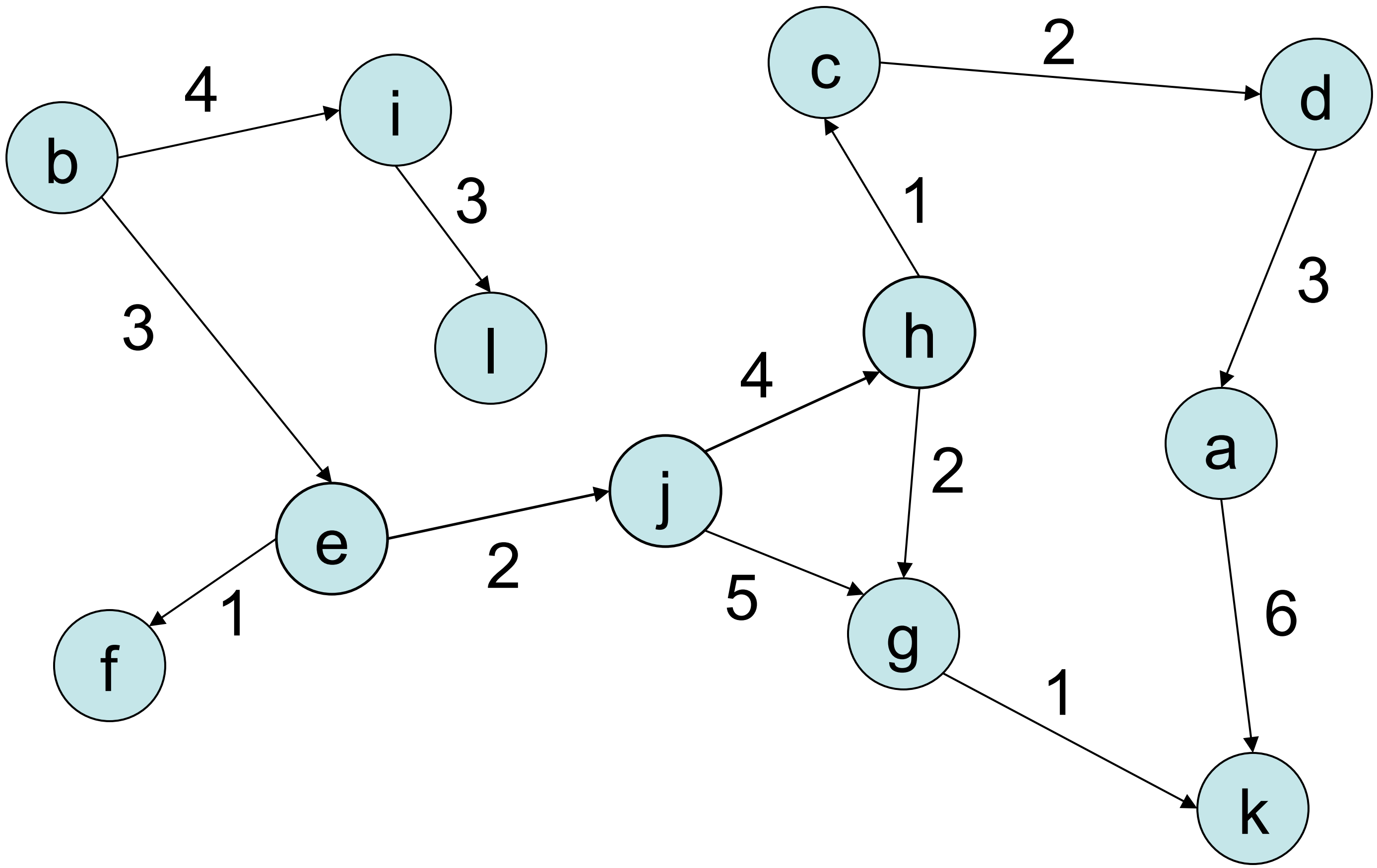
Informed search algorithms expand nodes according to a criteria.

Example: Best-first (greedy) search expands the nodes based on the cost of the edge.

Similar to DFS with the stack replaced by a priority queue (heap)

Best-First Search: example

Root at b, goal is a



A* Search

A classic algorithm that can ensure optimality and completeness is called A-star (A*).

A* uses a heuristic to help select the next vertex to expand: $h(V)$ is the heuristic for vertex V.

To implement use Best-First Search with the priority $f(V) = g(V) + h(V)$, where g is the path cost from the root

Example: N-puzzle problem

Some other important graph algorithms and problems

Topological Sorting

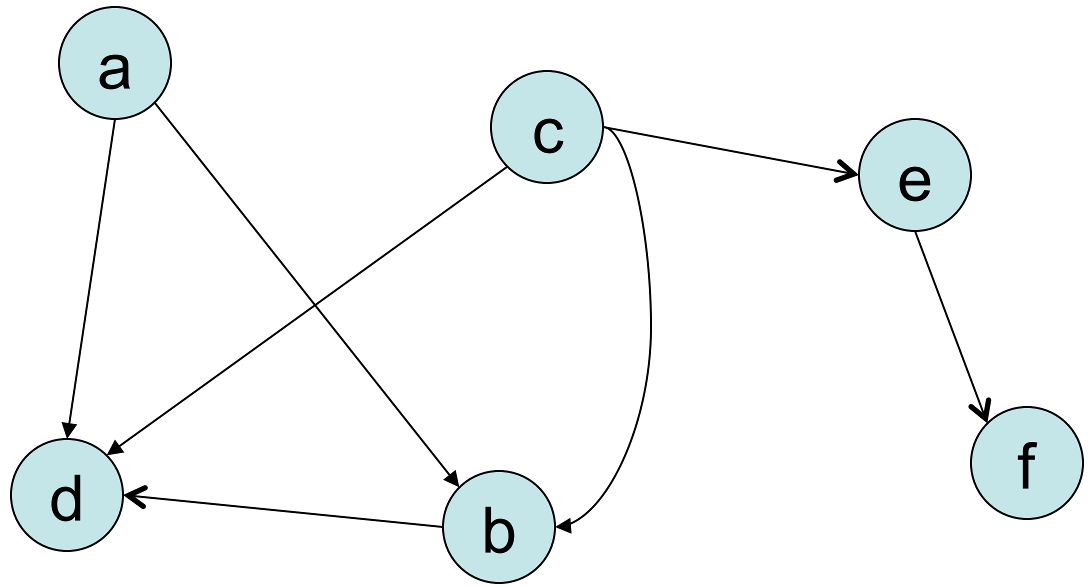
Minimal Spanning Tree

Shortest Path (Dijkstra's Algorithm), a simplification of A*

A very famous graph problem is the Traveling Salesperson Problem.

Example

Write a simple program to represent the graph below using an adjacency list.



After constructing the graph, print out all vertices connected to vertex a (or print none exist) using depth first search.

Next Actions and Reminders

Read CH pp. 671-681 on STL Containers

Program 5 is due 12/11.

Please fill out the SPOT survey!