Graphs

this is a flow chart, but
it is similar to a graph

Snarf today’s code

Today

• Our last topic
  • Intro to graphs
  • Coding with graphs
Graphs

- set of vertices
  - \{1, 2, 3, 4, 5, 6\}
- set of edges
  - \{(1, 2), (1, 4), (2, 5), (5, 3), (5, 6)\}

*directed graphs* – edge sets are ordered
- \{(1, 2), (1, 4), (2, 5), (5, 3), (5, 6)\}
- 1 points to 2 – notice the arrow
- (2, 1) is not an edge

*a.k.a. digraphs*
Graphs

• undirected graphs – edge sets are not ordered
  • \{(1, 2), (1, 4), (2, 5), (5, 3), (5, 6)\}
  • (1, 2) is the same as (2, 1)

Graphs

• edges can have weights
Graphs

• Why do you care?

facebook

Tabitha Peck

Graphs

• Kevin Bacon
Graphs

- Traveling salesperson problem
  - Given a list of cities and the distance between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the original city?

- Depth-first-search
  - explore as far as possible before backtracking

```
Start at root

def dfs(vertex):
    if visited vertex return;
    visit vertex
    for (adjacent vertices to vertex)
        dfs(adjacent vertex)
```
• Depth-first-search
  • explore as far as possible before backtracking

Start at root

\[
\text{dfs(vertex)}
\]
\[
\quad \text{if(visited vertex) return;}
\]
\[
\quad \text{visit vertex}
\]
\[
\quad \text{for(adjacent vertices to vertex)}
\]
\[
\quad \quad \text{dfs(adjacent vertex)}
\]
Graphs

- Depth-first-search
  - explore as far as possible before backtracking

Start at root

dfs(vertex)
  if(visited vertex) return;
  visit vertex
  for(adjacent vertices to vertex)
    dfs(adjacent vertex)

A B D

Graphs

- Depth-first-search
  - explore as far as possible before backtracking

Start at root

dfs(vertex)
  if(visited vertex) return;
  visit vertex
  for(adjacent vertices to vertex)
    dfs(adjacent vertex)

A B D F
• Depth-first-search
  • explore as far as possible before backtracking

Start at root

dfs(vertex)
  if(visited vertex) return;
  visit vertex
  for(adjacent vertices to vertex)
    dfs(adjacent vertex)

A B D F E
Graphs

- Depth-first-search
  - explore as far as possible before backtracking

Start at root

dfs(vertex)
  if(visited vertex) return;
  visit vertex
  for(adjacent vertices to vertex)
    dfs(adjacent vertex)

A B D F E C G

Graphs

- Breadth-first-search
  - explore as far as possible before backtracking

Start at root

bfs(vertex)
  myQ.enqueue(vertex)

while(!myQ.isEmpty())
  v = myQ.dequeue
  for(adj vertices of v)
    if(adj not visited)
      myQ.enqueue(adj)
Graphs

• Breadth-first-search
  • explore as far as possible before backtracking

Start at root

\[ \text{bfs(root) } \]
\[ \text{myQ.enqueue(root)} \]

while(!myQ.isEmpty())
  \[ v = \text{myQ.dequeue} \]
  \[ \text{for(adj vertices of v)} \]
    \[ \text{if(adj not visited)} \]
    \[ \text{myQ.enqueue(adj)} \]

A B C E D F G
Graphs

• Breadth-first-search
  • explore as far as possible before backtracking

Start at root

bfs(root)
  myQ.enqueue(root)

while(!myQ.isEmpty())
  v = myQ.dequeue
  for(adj vertices of v)
    if(adj not visited)
      myQ.enqueue(adj)

A B C E D F

A B C E D F G

Graphs

• Breadth-first-search
  • explore as far as possible before backtracking

Start at root

bfs(root)
  myQ.enqueue(root)

while(!myQ.isEmpty())
  v = myQ.dequeue
  for(adj vertices of v)
    if(adj not visited)
      myQ.enqueue(adj)

A B C E D F G
**Code time**

- snarf today’s code
  - this will be helpful for APT set 7

**Before you go**

- Today
  - Intro to graphs
  - Coding with graphs

- How are things going?
  - [http://goo.gl/lo1e5x](http://goo.gl/lo1e5x)
    - (lower-case L, lower-case O, #1, e, 5, x)