15.2
Connectivity

Connectivity

Given a graph $G = (V, E)$:

1. path: sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that $v_i v_{i+1} \in E$ for $1 \leq i \leq k-1$. The length of the path is $k-1$ (the number of edges in the path) and the path is from v_1 to v_k. Note: a single vertex u is a path of length 0.
2. cycle: sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that $\{v_i, v_{i+1}\} \in E$ for $1 \leq i \leq k-1$ and $\{v_1, v_k\} \in E$. Single vertex not a cycle according to this definition.
   Caveat: Some times people use the term cycle to also allow vertices to be repeated; we will use the term tour.
3. A vertex u is connected to v if there is a path from u to v.
4. The connected component of u, $\text{con}(u)$, is the set of all vertices connected to u. Is $u \in \text{con}(u)$?

Connectivity

Given a graph $G = (V, E)$:

1. path: sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that $v_i v_{i+1} \in E$ for $1 \leq i \leq k-1$. The length of the path is $k-1$ (the number of edges in the path) and the path is from v_1 to v_k. Note: a single vertex u is a path of length 0.
2. cycle: sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that $\{v_i, v_{i+1}\} \in E$ for $1 \leq i \leq k-1$ and $\{v_1, v_k\} \in E$. Single vertex not a cycle according to this definition.

   Caveat: Some times people use the term cycle to also allow vertices to be repeated; we will use the term tour.
3. A vertex u is connected to v if there is a path from u to v.
4. The connected component of u, $\text{con}(u)$, is the set of all vertices connected to u. Is $u \in \text{con}(u)$?

Connectivity

Given a graph $G = (V, E)$:

1. path: sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that $v_i v_{i+1} \in E$ for $1 \leq i \leq k-1$. The length of the path is $k-1$ (the number of edges in the path) and the path is from v_1 to v_k. Note: a single vertex u is a path of length 0.
2. cycle: sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that $\{v_i, v_{i+1}\} \in E$ for $1 \leq i \leq k-1$ and $\{v_1, v_k\} \in E$. Single vertex not a cycle according to this definition.

   Caveat: Some times people use the term cycle to also allow vertices to be repeated; we will use the term tour.
3. A vertex u is connected to v if there is a path from u to v.
4. The connected component of u, $\text{con}(u)$, is the set of all vertices connected to u. Is $u \in \text{con}(u)$?

Connectivity

Given a graph $G = (V, E)$:

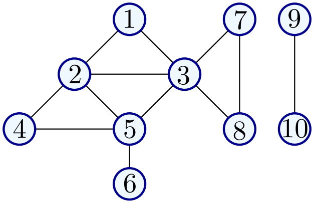
1. path: sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that $v_i v_{i+1} \in E$ for $1 \leq i \leq k-1$. The length of the path is $k-1$ (the number of edges in the path) and the path is from v_1 to v_k. Note: a single vertex u is a path of length 0.
2. cycle: sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that $\{v_i, v_{i+1}\} \in E$ for $1 \leq i \leq k-1$ and $\{v_1, v_k\} \in E$. Single vertex not a cycle according to this definition.

   Caveat: Some times people use the term cycle to also allow vertices to be repeated; we will use the term tour.
3. A vertex u is connected to v if there is a path from u to v.
4. The connected component of u, $\text{con}(u)$, is the set of all vertices connected to u. Is $u \in \text{con}(u)$?

Connectivity contd

Define a relation C on $V \times V$ as uCv if u is connected to v

1. In undirected graphs, connectivity is a reflexive, symmetric, and transitive relation. Connected components are the equivalence classes.
2. Graph is connected if there is only one connected component.



Connectivity Problems

Algorithmic Problems

1. Given graph G and nodes u and v, is u connected to v?
2. Given G and node u, find all nodes that are connected to u.
3. Find all connected components of G.

Can be accomplished in $O(m + n)$ time using **BFS** or **DFS**.
BFS and **DFS** are refinements of a basic search procedure which is good to understand on its own.

Connectivity Problems

Algorithmic Problems

1. Given graph G and nodes u and v, is u connected to v?
2. Given G and node u, find all nodes that are connected to u.
3. Find all connected components of G.

Can be accomplished in $O(m + n)$ time using **BFS** or **DFS**.
BFS and **DFS** are refinements of a basic search procedure which is good to understand on its own.

THE END

...

(for now)