Graphs and Algorithms

January 7, 2019

---

1 Graphs

- A graph is a $pair(v, e)$ where v is a set of nodes called *vertices* and e is a collection of pairs of vertices, called *edges*
- Vertices and Edges are positions and store elements
- insert graph example here

---

2 Edge Types

1. Directed Edge
   - Ordered pair of vertices(u, v)
   - First vertex u is the origin
   - Second vertex v is the destination
   - e.g. a flight
2. Undirected Edge
   - Unordered pair of vertices(u, v)
   - e.g. a flight route
3. Directed Path
   - All edges are directed
   - e.g. a route network
4. Undirected Path
   - All the edges are undirected
   - e.g. a flight network

---

3 Subgraphs

- A ***subgraph*** S of a graph G is such that
  - The vertices of S are a subset of the vertices of G
  - The edges of S are a subset of the edges of G
- A ***spanning subgraph*** of G is a subgraph that contains all the vertices of G, but not necessarily all the edges of G

---

4 Connectivity

- A graph is considered ***Connected*** if there is a path between every pair of vertices
- A ***Connected Component*** of a graph G is a maximal connected subgraph of G

---

5 Trees and Forests

- A ($free$) ***tree*** is an undirected graph T such that
  - T is connected
  - T has no cycles
  - Note that a free tree is different from a rooted tree
- A ***forest*** is an undirected graph without cycles
  - The connected components of a forest are trees
- A ***spanning tree*** of a connected graph is a spanning subgraph that is a tree
  - Is not unique unless the graph is a tree
- A ***spanning forest*** of a graph is a spanning subgraph that is a forest

---

6 Depth-First Search

- Depth-First Search (DFS) is a general technique for traversing a graph
- A DFS traversal of a graph G
  - Visits all the vertices and edges of G
  - Determines whether G is connected
  - Computes the connected components of G
  - Computes a Spanning Forest of G