CS/ECE 374 FALL 2018	Zhe Zhang (zzhan157@illinois.edu)
Homework 7 Problem 3	Ray Ying (xinruiy2@illinois.edu)
	Anqi Yao (anqiyao2@illinois.edu)

Graphs are a powerful tool to model many phenomena. The edges of a graph model pairwise relationships. It is natural to consider higher order relationships. Indeed *hypergraphs* provide one such modeling tool. A hypergraph $G = (V, \mathcal{E})$ consists of a finite set of nodes/vertices V and a finite set of hyper-edges $\mathcal{E}$. A hyperedge $e \in \mathcal{E}$ is simply a subset of nodes and the cardinality of the subset can be larger than two. An undirected graph is a hypergraph where each hyper-edge is of size exactly two. Here is an example. $V = \{1, 2, 3, 4, 5\}$ and $\mathcal{E} = \{\{1, 2\}, \{2, 3, 4\}, \{1, 3, 4, 5\}, \{2, 5\}\}$. The representation size of a hypergraph is $|V| + \sum_{e \in \mathcal{E}} |e|$. An alternating sequence of nodes and edges $x_1, e_1, x_2, e_2, \ldots, e_{k-1}, x_k$ where $x_i \in V$ for $1 \leq i \leq k$ and $e_j \in \mathcal{E}$ for $1 \leq j \leq k-1$ is called a path from u to v if (i) $x_1 = u$ and $x_k = v$ and (ii) for $1 \leq j < k$, $x_j \in e_j$ and $x_{j+1} \in e_j$.

- Given a hypergraph $G = (V, \mathcal{E})$ and two nodes $u, v \in V$ we say that u is connected to v if there is path from u to v. We say that a hypergraph is connected if each pair of nodes u, v in G are connected. Describe an algorithm that given a hypergraph G checks whether G is connected in linear time. In essense describe a reduction of this problem to the standard graph connectivity problem. You need to prove the correctness of your algorithm.
- Suppose we want to quickly spread a message from one person to another person during an emergency on a social network called AppsWhat which is organized as a collection of groups. Messages sent by a group member are broadcast to the entire group. AppsWhat knows the members and the list of groups on its service. The goal is to find the fewest messages that need to be sent such that a person u can reach a person v. Model this problem using hypergraphs and describe a linear-time algorithm for it. No proof necessary for this part.

Solution:

1. We can use a single additional vertex represent a hyperedge in the graph G. Hence, We can construct a indirect graph G' to convert the hypergraph $G = (V, \mathcal{E})$ into a standard graph where graph $G' = (V', E')$:
   $V' = V \cap \mathcal{E}$
   $E' = \{(v, e) | v \in V, e \in \mathcal{E}, v \in e\}$
   The running time construct the graph G' will be $O(|V| + |\mathcal{E}| + \sum_{e \in \mathcal{E}} |e|)$, which is in linear-time.
   After constructing the graph G', we can start at an arbitrary vertex in the graph G' and using DFS algorithm to find the total number of vertex it connects to. We can compare it to $|V|$ afterwards. If it's the same, return true. Otherwise, false.
   The proof for this algorithm:
   From the definition of hypergraph, An alternating sequence of nodes and edges

$$x_1, e_1, x_2, e_2, \ldots, e_{k-1}, x_k$$

where $x_i \in V$ for $1 \leq i \leq k$ and $e_j \in \mathscr{E}$ for $1 \leq j \leq k-1$ is called a path from u to v if (i) $x_1 = u$ and $x_k = v$ and (ii) for $1 \leq j < k$, $x_j \in e_j$ and $x_{j+1} \in e_j$. By the definition of G', e_j in the old path is (v_j, e_j) and there is vertex for each edge as well, each e_j will be a vertex representing an edge. So the path for u to v in G' will be

$$x_1, (x_1, e_1), e_1, (x_2, e_1), x_2, (x_2, e_2), e_2, ..., x_k$$

Also, when giving the path from u to v in G' in the form of

$$x_1, (x_1, e_1), e_1, (x_2, e_1), x_2, (x_2, e_2), e_2, ..., x_k$$

To transfer to the path in G, since (x_j, e_j) are edges in G' and e_1 are not vertex but edge, so the path from u to v in graph G can only be

$$x_1, e_1, x_2, e_2, \ldots, e_{k-1}, x_k$$

. Hence, we proved the correctness of the algorithm. The running time for the algorithm has the constructing time and DFS time. Both will be in linear time as we just discussed that constructing time is linear, so the total time complexity will be $O(|V| + \sum_{e \in \mathscr{E}} |e|)$.

2. So basically, we want to think a hypergraph G with each person as a vertex and each group as a hyperedge. So the question for the fewest message for person u to reach person v will be finding the shortest path between u and v in the hypergraph G. Then we can construct a graph G' the same way in part 1 for the hypergraph G. We will use a BFS algorithm on it to find the shortest path between u and v. After we find the value k, we can divided k in half. Since as we constructed in part 1, one edge in hypergraph will be transformed into 2 edges. So the fewest message needed to be sent is $k/2$. The running time for this algorithm is $O(|V| + \sum_{e \in \mathscr{E}} |e|)$ since both constructing graph and BFS are in linear time.

■