Algorithms Homework #2
Spring 2017

A magnitude 8 earthquake hits a large city. EMS teams and hospitals are located throughout the city. As they respond to emergency calls, they discover roads that have been destroyed. Repair teams are periodically allocated to repair roads.

Data Structures:

1. Roads are edges
2. Street addresses are vertices.
    a. An address will include the following information: number and street name. For example: "221B Baker Street"
    b. Addresses should be grouped into compound vertices, i.e. a sorted array of addresses; binary search gives precise location of an address within a group. Long stretches of addresses on a street can be split into multiple compound vertices.
    c. At an intersection, the four addresses at the corners of the intersection should all have edges to each other.
3. People, hospitals, and EMS units are not vertices, rather they are objects that get associated with the address/vertex at which they are located.
4. All people who call for help are placed in a single priority queue. Life threatening emergencies get a higher priority, etc.
    a. In theory, a person could have their priority raised after a given time based on a medical algorithm that calculates how long the person with a given condition can wait before they become more critical, but we are ignoring this for purposes of this assignment and accepting the possibility of starvation

Dispatch Algorithm

- Assuming:
    1. there are *k* EMS teams
    2. hospitals are distributed evenly throughout the city
- Run BFS from newly available EMS team to first *k* patients in the patient queue
    - this implies the need for the queue implementation to provide a method to return first *k* items without dequeueing them
    - this must be implemented without the need to move any entries around in the queue; it must be O(1), i.e. a constant cost, to "peek" at the first *k* items
- Send the newly available EMS team to the patient from that subset of patients with the shortest path from the EMS team's current location, and remove that patient from the patient queue
- Once an EMS unit is assigned to a patient, that unit can't be assigned to another patient until the other EMS units have treated *k*-1 patients.

Broken Road Discover, Mark, and Repair

- ***Discover***: roads are only discovered to be broken when an EMS unit follows a path that contains the broken road. When a road is discovered to be broken:
    - Mark the road, as described below

    - Return the patient assigned to the EMS unit to the patient priority queue and run a new BFS for the EMS unit from its current location to find the closest patient amongst the *k* first patients in the queue
- ***Mark:*** Mark the edges representing the broken road as unpassable
    - Do not remove the edge(s) from the graph – we will want to repair it
    - If a broken road segment passed through an intersection, that intersection is unpassable/broken in all directions
- ***Repair:*** DFS to detect connected components.
    - If >1 component:
        - Count patients and teams that are found within each component. Sort components based on the number of EMS units and hospitals they contain.
        - Do BFS from component with smallest number of EMS units and hospitals to find nearest EMS units and hospital, recording each broken road you'd have to use to get there.
        - Repair the roads needed to connect the given component to the nearest EMS units and hospital. If reaching an EMS unit and reaching a hospital require repairing the same number, but different, roads, connecting to a hospital takes priority over connecting to EMS
        - Assume that repairing a road takes the same amount of time it takes for 2*k* people to be treated by EMS
        - Once a component becomes reconnected to at least one EMS Unit or hospital, loop back to the beginning of the "repair" algorithm
    - If 1 component: multisource BFS from hospitals to broken roads. Broken roads closest to hospitals get fixed first.

Input File Description

Your program will take a path to an input file as a command line argument. The file will be of the following form (ABNF):

```
input = 1*address *edge 1*intersection 1*hospital 1*EMS_Unit 1*event

address = ID house_number street_name ; indicates addresses in your
city at which people, EMS units, or hospitals can be located
ID = 1*DIGIT ; an ID number used for an address in the remainder of
the input
house_number = 1*DIGIT
street_name = 1*VCHAR
edge = ID "," ID ; the two addresses with the specified IDs are
connected by an edge. Note that two addresses with the same
street_name and consecutive house_numbers (e.g. 2495 Amsterdam Ave.
and 2496 Amsterdam Ave.) must automatically have an edge between them
without it being specified in the input. "edge" will be used to define
edges when streets change names.

intersection = ID "," ID "," ID "," ID ; the IDs of the 4 addresses at
the 4 corners of the intersection
```

```
hospital = "hospital" ID ; indicates that a hospital is located at the address with the given ID

EMS_Unit = "EMS" ID ; indicates that an EMS unit is housed at the address with the given ID

event = 911CallGroup / 911call / brokenRoad / repairTeam
911CallGroup = "Being911CallGroup\n" 1*911call "\nEnd911CallGroup"
911call = house_number street_name severity
severity = 1*DIGIT ; higher numbers are more severe
brokenRoad = house_number house_number street_name ;the two house_numbers are the start and end of the segment of the given street that is broken. If that segment passes through an intersection, that intersection in unpassable
repairTeam = "repair team" ; indicates that one repair team is available to you
```

Building the Graph from the Input

The first event will appear in the file only after all other information has been given