**Routing in Wireless Ad-hoc Networks**

**Warning**: you must not attempt to get help from external sources such as online sites that offer solutions or obtain code from other students. Any submission for grading must be your own work. You must not copy from others or allow your work to be copied. We will check for similarities of code. Any violation will result in your receiving zero for this project or an F for this course. No exception.

The lab submission must include a well-written report. Your report must adequately describe your solutions, such as algorithm design, pseudo code, time complexity etc.

**Total points: 100**

This lab must be completed using Java and is an individual project. Naming of your programs must strictly follow the following convention: (1) one file for each part; (2) Lab1 followed by your Uid and I for part I (for example Lab1U00998499I.java). Upload only your source code (one file for each part), and a report to the Pilot dropbox.

**Note**: The location of the input file and the output file must be your local directory. In your programs, a directory or a path MUST NOT be used for your input/output files. Use only a file name, i.e, no absolute path.

The solution time complexity for this lab is an important consideration in grading. Therefore, you should design and implement your solutions carefully. Test cases will exercise your programs thoroughly. Your programs should efficiently handle potentially large problems. Your code must produce correct solutions for all test cases within 4-5 seconds.

**Grading**. You will be given only 1 simple case in the problem description below. Your solutions will be tested with more cases that are of much larger size. These test cases will NOT be provided to you. In real-world, you never know what the test cases are. Test cases will exercise your programs thoroughly. Therefore, your programs should efficiently handle potentially large problems. Your lab will be graded based on how many test cases you can pass. There will be no partial credit for a case. Therefore, you should devise more cases on your own to test your code. You need to consider border cases and cases of very large size before submitting your lab.

**Problem** An ad-hoc wireless network consists of numerous nodes. In the lecture, we discuss that we can model the wireless communication channel in different ways, such as point-to-point model or broadcast channel model.

In this lab, you will be asked to implement two algorithms to construct a broadcast tree rooted at a given source node in the network.

**Part I. Broadcast Incremental Power (BIP)**

In Part I, you are asked to write a program to implement the BIP algorithm for constructing a broadcast tree and determine the units of power needed. In this part, the BIP algorithm takes advantage of the broadcast nature (i.e., assume the broadcast channel model) of the communication channel.

**INPUT FORMAT (file adhoc1.txt):**

The first line of input specifies the source node, the number of nodes, N (where 1≤N≤100,000), and M pairs of nodes. Each of the next M lines contains three integers a, b, and p that describe transmission power between two nodes a and b where p units of power are needed for one node to reach the other (assuming symmetrical channel).

**OUTPUT FORMAT (file adhoc1out.txt):**Your solution must produce a single integer to indicate the total units of power needed by the BIP algorithm to construct a broadcast tree that reaches all nodes of the ad-hoc wireless network.  **SAMPLE INPUT:**

1 5 7

1 2 5

1 3 1

1 4 3

1 5 10

2 3 3

3 4 7

4 5 1

**SAMPLE OUTPUT:**

6

**Part II. Minimum Spanning Tree**

In Part II, we consider the same ad-hoc wireless network, but assume point-to-point communication channel. You will implement the minimum spanning tree algorithm (the Prim’s algorithm) to construct a broadcast tree.

**INPUT FORMAT (file adhoc2.txt):**

The input format is the same as that of Part I.

**OUTPUT FORMAT (file adhoc2out.txt):**

Your solution must produce a single integer to indicate the total units of power needed by the minimum spanning tree algorithm to construct a broadcast tree that reaches all nodes of the ad-hoc wireless network.

**SAMPLE INPUT:**

1 5 7

1 2 5

1 3 1

1 4 3

1 5 10

2 3 3

3 4 7

4 5 1

**SAMPLE OUTPUT:**

8