Data Structures Assignment 3: Kruskal's Algorithm (and Prim's)

Instructor: Benjamin Geiger (`begeiger@mail.usf.edu`)

Assigned 3 July 2014 — Due 17 July 2014, 9:30 AM

Assignment

Download the `KruskalProject.tar.gz` file from Canvas and extract it. Inside, you'll find several source files. The `Kruskal.cpp` file is incomplete; you are to complete the `kruskal()` function.

Extra Credit (10 pts): `Prim.cpp` is also incomplete; you are to complete the `prim()` function. Be warned that this is more difficult than Kruskal's algorithm. There is information available in a slide set on Canvas, but it will require a significant amount of effort.

Details

The program in `Kruskal.cpp` is to implement Kruskal's algorithm. (*Mutatis mutandis* for `Prim.cpp`.)

The `Edge` class is implemented for you. You can create an edge with the syntax `Edge(from, to, weight)`.

Likewise, the Union-Find algorithm is implemented for you in `UnionFind.cpp`. Note that instead of multiple `makeset()` calls, you simply create a single `UnionFind` object with a specified size n; it automatically creates single-element sets for 0 through $n - 1$.

Instructions

To compile your program in Linux, simply type `make` at the command prompt. The included `Makefile` should handle the rest.

Note: To compile `Prim.cpp`, type `make Prim` at the command prompt.

A sample input file (`input0`) and output file (`correctK0`) have been provided. To test your code:

1. Compile your code, as above.
2. Enter at the command prompt:
   `./Kruskal < input0 > testK0`

   This runs the program and tells it to accept input from `input0` and save output to `test0`.
3. Compare the output to the correct output:
   `diff -u testK0 correctK0`

   Lines where your output differs from the correct output will be printed; your lines will be prefaced with `-` and the correct lines will be prefaced with `+`. (Lines with no characters are common to both and are included for context.)

Administrivia

- Include your name in the comment at the beginning of the `Kruskal.cpp` file (and `Prim.cpp` if you choose to complete it).
- Submit only `Kruskal.cpp` (and `Prim.cpp` if you choose to complete it). The other files should not be modified; they will be replaced with pristine versions before grading.
- Your code will be tested on a Linux system; other Unix systems are likely to work properly, but Windows (specifically Visual Studio) often causes problems. If in doubt, test your code on the C4 lab PCs or on CIRCE.

  Submissions that do not compile will be given a grade of 20%.
- Your code will be tested on several input files other than those provided.
- Please properly indent your code. Most code editors contain an automatic indentation feature.
- Late submissions will be penalized 10% for each day (or fraction thereof) late, to a maximum of 3 days.