ECS 60 **Programming Assignment #5 (50 points, 1 hour)** **Spring 2015**

Due: **Tuesday**, June 2nd, at 11:59 pm to p5 in the cs60 account.
Filenames: router.cpp, router.h, Makefile, authors.txt. Executable name: router.out

authors.txt should be in the format:
First line: <e-mail of first partner> <space> <name of first partner, last name first then comma then first name>
Second line (if needed): <e-mail of second partner> <space> <name of second partner in above format>

The energy regulators of United States must determine how best to transfer the electricity from tens of thousands of power plants to those areas that need it. You are to write a program that will determine how best to transfer electricity from one city to another to minimize the amount of electricity transferred within the country. The program takes a power filename as its only parameter. You will find a barebones Makefile, RouterRunner.h, RouterRunner.cpp, barebones router.cpp, barebones router.h, my router.out, data files, CreatePowerFile.out, and CreatePowerFile.cpp in ~ssdavis/60/p5.

1. Power Files:
   1.1. The power files are read by the RouterRunner.cpp file that I have already written so you need not worry about reading it.
   1.2. The file is created by CreatePowerFile.out, which is compiled from CreatePowerFile.cpp.
   1.3. The names of the files indicate the number of cities, and the seed used for the random number generator.
   1.4. The first line of a file states the number of counties.
   1.5. Each succeeding line of the file holds the information for one city. With information in the following order: City number, electrical usage, electrical production, and a list of up to eight adjacent cities.
   1.6. There is exactly as much usage as there is production in the country.
2. Solutions:
   2.1. Your setTransfers() function fills in the transfers matrix with the number electrical units transferred from each city to its adjacent cities. Note that the second index order is up to you. So your program must set the destCity value in each transfer rather rely on it corresponding to the destination in the adjList of the CityInfo. Nonetheless, there are only eight possible transfers from each city, so you are expected to combine all transfers from one city to one of its adjacent cities.
3. Grading:
   3.1. (25 points) The Total time of a team's program on 3 power files with 25000 cities, compared with the time of my program. Time score = min (30, 25 * Sean's Total CPU Time / Your Total CPU Time).
   3.2. (25 points) The Total of a team's transfers on the 3 power files compared with the transfers of my program,
      3.2.1. Moving one unit from one city to an adjacent city increments the total. If a unit of electricity has to move across four three cities then it will count as 3 transfers units, i.e the path length. If two units move between adjacent cities, then that would count as 2 transfer units.
      3.2.2. Transfer score = min (30, 25 * Sean's Total Transfers / Your Total Transfers).
   3.3. If the checkFlows() function in RouterRunner.cpp generates any error message, then the program receives a zero.
      3.3.1. The errors detected are: 1) transferring between two cities that are not adjacent; 2) a city not having exactly as much electricity as needed, and 3) the total provided does not match the total calculated.
   3.4. Programs must be compiled without any optimization options. You may not use any precompiled code, including the STL and assembly. Other than Weiss files, you must have written all of the code you submit.
   3.5. You may not have any static, or global variables since they would be created before the CPU timer begins.
   3.6. Since I have a short time to grade these, if your CPU time exceeds 30 seconds for a power file, then please send me an e-mail when you submit it so I can run it separately. If you do not send me such a warning, I will deduct 5 points from your score.
4. Miscellaneous:
   4.1. Brainstorm a lot before starting to code. Think about which operations are done a lot, and minimize their cost.
   4.2. Even though the TAs will not see my solution, I have instructed the TAs to not help you with your design. They will help with debugging though.
   4.3. While it only took me an hour to write this, I spent five hours debugging only to find out, I needed to change one "short" to "int"!!!
   4.4. Keep things simple, and get things running first, and only then use gprof to learn where things are going slowly.
   4.5. Remember to turn in dsexceptions.h if your program needs it!

4.6. If you find you have a bug in the middle of running through a large power file, add a an if statement in your code that describes the state of the machine at the time of the bug, and then put a breakpoint at the cout in the if statement. For example: if(city == 978 && adjPos == 4) cout << "Help!\n";

4.7. Finally, there will not be a gold version of this program. I originally wrote this challenge in 2003. Though I did not look at that old code, the version I wrote this time is faster and uses fewer transfers than the gold version from 2003!!

```
typedef struct
{
  int adjList[8];
  short adjCount;
  int usage;
  int production;
} CityInfo;


class Transfer
{
public:
  int destCity;
  unsigned amount;
  Transfer(): destCity(0), amount(0) {}
}; // class Transfer


int main(int argc, char** argv)
{
  CPUTimer ct;
  int numCities;
  Transfer **transfers = new Transfer*[25000];
  for(int i = 0; i < 25000; i++)
    transfers[i] = new Transfer[8];
  CityInfo *infos = readFile(argv[1], &numCities);
  CityInfo *infos2 = new CityInfo[numCities];
  memcpy(infos2, infos, numCities * sizeof(CityInfo));
  ct.reset();
  Router *router = new Router(infos, numCities);
  delete [] infos;
  int theirTotal = router->setTransfers(transfers);
  double time = ct.cur_CPUTime();
  checkFlows(infos2, numCities, transfers, theirTotal);
  cout << "Transfer total: "  << theirTotal << " CPU Time: " << time << endl;
  return 0;
} // main())
```

```
[ssdavis@lect1 p5]$ cat power5_5.csv
5
0,5632,1920,1,2,3
1,649,9073,0,4,2,3
4,4381,1026,1,3
2,6848,5860,0,1,3
3,369,0,0,1,4,2
 [ssdavis@lect1 p5]$ router.out power5_5.csv
Transfer total: 8424 CPU Time: 0.004626
[ssdavis@lect1 p5]$ router.out power25000_2.csv
Transfer total: 135421588 CPU Time: 0.41753
[ssdavis@lect1 p5]$
```