ECS 60 **Programming Assignment #5, 50 points (2 hours)** **Fall 2015**

Due **Thursday**, December 3rd, 11:59pm. Executable name: RunCity.out
Files NOT to handin to cs60 p5 are: RunCity.cpp, RunCity.h, CPUTimer.h
Minimum files to handin to cs60 p5 are: Makefile, StreetSelecter.h, StreetSelecter.cpp, authors.csv.

Format of authors.csv: author1_email,author1_last_name,author1_first_name
author2_email,author2_last_name,author2_first_name
For example: simpson@ucdavis.edu,Simpson,Homer
potter@ucdavis.edu,Potter,Harry

You are to write an efficient program that will determine which 10 miles of roads should be widened in a city to minimize the time for 10,000 automobile trips. When a road is widened its maximum speed increases from 25 MPH to 35 MPH. I have provided the driver program RunCity.cpp, and the necessary class stubs in StreetSelecter.h and StreetSelecter.cpp. You may add any classes and/or code you wish to StreetSelecter.cpp, and StreetSelecter.h. You may also use, and alter any Weiss files. I have written a program, named ShowCity.out that will display the roads of a city. The program RunTrips.out tests the solutions provided by your program to determine how much faster the 10,000 trips take. RunTrips.out is called from RunCity.out. You are also welcome to look at CreateCity.cpp that is the source for CreateCity.out that creates the data files. All files mentioned, as well as my executable, RunCity.out, can be found in ~ssdavis/60/p5.

Further specifications:

1. Command Line parameter is the name of the City file.
2. City Files
   - 2.1. Since RunCity.cpp handles all file input, you need not concerned with how to read City files and ans files.
   - 2.2. City file names are appended with four numbers: number of tenth of miles on a side of the City, number of streets, number of trips, and the seed used for the random number generator.
   - 2.3. Each unit of measurement is one tenth of a mile.
   - 2.4. Streets are from two tenths to fourteen tenths in length. All streets are connected to each other.
   - 2.5. The coordinate system starts with (0,0) being in the Northwest corner.
   - 2.6. Each end of a Trip is an intersection of two to four streets.
   - 2.7. Format
     - 2.7.1. First line is the width of the city in tenths of miles.
     - 2.7.2. Next set of lines until a -1 -1 are Block descriptions with the format:
       <Start NS coordinate> <StartEW coordinate> <4 road numbers from start vertex in NESW order> <End NS coordinate> <End EW coordinate> <4 road numbers from the end vertex in NESW order><Length>
     - 2.7.3. Last set of lines are the trips with the format:
       <Start NS coordinate> <StartEW coordinate> <End NS coordinate> <End EW coordinate>
3. Measurements
   - 3.1. CPU time starts just before constructing your Database object in main(), and ends after the last query() is handled. Thus, your destructors will not be called during CPU time.
     - 3.1.1. You may not have any global variables since they would be constructed before CPU time starts.
   - 3.2. "Time difference" is the difference of running the shortest (by time) paths of the 10,000 trips with all streets having a speed limit of 25 MPH, running the shortest (by time) paths of the 10,000 trips with the speed limits of the 10 miles of streets indicated in the widenedBlocks array increased to 35 MPH. Since blocks may be longer than one tenth of a mile, the number of entries your program places in the widenedBlocks array may well be less than 100. It is up to your program to set numWidened to the appropriate value. The .ans files are copies of widenedBlocks.
4. Grading
   - 4.1. The program will be tested using three 150 city size, 1000 street, 10000 trip files. The measurements will be the total of the three runs.
   - 4.2. If there are ANY error messages from RunCity, then the program will receive zero.
   - 4.3. If your code has no error messages, and takes less than 30 CPU time, then you will receive at least 30 points.
   - 4.4. CPU Time score = min(13, 10 * Sean's CPU / Your CPU)
   - 4.5. Time Difference score = min(13, 10 * Your Time Difference / Sean's Time Difference)
5. Makefile. The Makefile provided contains the minimum needed to create RunCity.out.
   - 5.1. You may not use an optimization flags in your Makefile.
   - 5.2. All code must be C++ source code.

5.3. To ensure that you handin all of the files necessary for your program, you should create a temp directory, and copy only *.cpp, *.h, and Makefile into it. Then try to make the program. Many students have forgotten to handin dsexceptions.h.

6. Suggestions.
   6.1. Start early, and get something working without errors.
   6.2. Don't fuss about speed until you have something working. Too many students fail to have a correct program by the deadline because they spend too much time tweaking early on.
      6.2.1. You will learn a lot just getting it running. Then you can use this knowledge to improve your code.
   6.3. Leave lots of time for testing and debugging. Test with all of the available files.
   6.4. Hint: The order of processing the trips may effect your CPU time.

```
[ssdavis@lect1 p5]$ RunCity.out
City150-1000-10000-1.txt
CPU Time: 1.2723
Original distance: 54028.60 miles
Time: 2161.14 hours  Speed: 25.00 MPH
Widened distance:  54186.20 miles
Time: 2071.77 hours  Speed: 26.15 MPH
Time difference: 89.38
[ssdavis@lect1 p5]$

[ssdavis@lect1 p5]$ cat RunCity.h
enum {NORTH, EAST, SOUTH, WEST};
enum {NS, EW};

class Block
{
public:
  short startCoordinates[2]; // NS
first, EW second
  short startRoads[4];  // road #s
leaving start, ordered NESW
  short endCoordinates[2]; // NS
first, EW second
  short endRoads[4];  // roads #s
leaving end, ordered NESW
  short length;
}; // Block class

class Trip
{
public:
  short startCoordinates[2]; // NS
first, EW second
  short endCoordinates[2];  // NS
first, EW second
}; // Trip class
[ssdavis@lect1 p5]$
```

```
[ssdavis@lect1 p5]$ cat City35-10-10-2.txt
35
6 21 0 2 0 0 6 23 0 2 3 2 2
6 23 0 2 3 2 6 26 0 0 0 2 3
7 15 0 10 0 0 7 19 6 10 6 10 4
7 19 6 10 6 10 7 20 4 10 4 10 1
7 20 4 10 4 10 7 23 3 10 3 10 3
7 23 3 10 3 10 7 25 0 0 0 10 2
9 16 0 5 0 0 9 17 0 5 9 5 1
9 17 0 5 9 5 9 19 6 5 6 5 2
9 19 6 5 6 5 9 20 4 5 4 5 1
9 20 4 5 4 5 9 23 3 5 3 5 3
9 23 3 5 3 5 9 25 0 0 0 5 2
11 13 0 7 0 0 11 17 9 7 9 7 4
11 17 9 7 9 7 11 19 6 7 8 7 2
11 19 6 7 8 7 11 23 3 0 3 7 4
14 11 0 1 0 0 14 17 9 1 9 1 6
14 17 9 1 9 1 14 19 8 1 8 1 2
14 19 8 1 8 1 14 20 0 0 0 1 1
9 17 0 5 9 5 11 17 9 7 9 7 2
11 17 9 7 9 7 14 17 9 1 9 1 3
14 17 9 1 9 1 17 17 9 0 0 0 3
6 19 0 0 6 0 7 19 6 10 6 10 1
7 19 6 10 6 10 9 19 6 5 6 5 2
9 19 6 5 6 5 11 19 6 7 8 7 2
11 19 6 7 8 7 14 19 8 1 8 1 3
14 19 8 1 8 1 17 19 8 0 0 0 3
2 20 0 0 4 0 7 20 4 10 4 10 5
7 20 4 10 4 10 9 20 4 5 4 5 2
9 20 4 5 4 5 10 20 4 0 0 0 1
6 23 0 2 3 2 7 23 3 10 3 10 1
7 23 3 10 3 10 9 23 3 5 3 5 2
9 23 3 5 3 5 11 23 3 0 3 7 2
11 23 3 0 3 7 13 23 3 0 0 0 2
-1 -1
7 19 14 17
9 23 11 17
9 19 7 23
7 19 14 19
7 19 9 17
11 19 11 17
6 23 7 20
6 23 14 19
9 19 7 20
14 19 11 19
[ssdavis@lect1 p5]$

[ssdavis@lect1 p5]$ ShowCity.out City35-10-10-2.txt

  0 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
  0

  1

  2                                          |
                                             |
  3                                          |
                                             |
  4                                          |
                                             |
  5                                          |
                                             |
  6                                        | | ----+-----
                                           | |     |
  7                                  --------+-+-----+---
                                           | |     |
  8                                        | |     |
                                           | |     |
  9                                  --+---+-+-----+---
                                       |   | |     |
h=left, j=down, k=up, l=right, caps = paging, x = exit
[ssdavis@lect1 p5]$ RunCity.out City35-10-10-2.txt
CPU Time: 0.000421
Original distance: 5.80 miles  Time: 0.23 hours  Speed: 25.00 MPH
Widened distance:  5.80 miles  Time: 0.19 hours  Speed: 30.39 MPH
Time difference: 0.04
[ssdavis@lect1 p5]$ head City35-10-10-2.ans
7 19 9 19
9 19 9 20
9 17 9 19
11 19 14 19
9 17 11 17
6 23 7 23
7 20 7 23
7 20 7 23
11 17 11 19
11 19 11 23
```

```
int main(int argc, char* argv[])
{
  Block *blocks;
  StreetSelecter *streetSelecter;
  int citySize, blockCount, numTrips, numWidened;
  Trip *trips, widenedBlocks[100];
  char filename[80], command[80];
  CPUTimer ct;

  if(argc != 2)
  {
    cout << "usage: RunCity.out CityFilename\n";
    return 1;
  }

  ifstream inf(argv[1]);
  strcpy(filename, argv[1]);
  strtok(filename, "-");
  strtok(NULL, "-");
  numTrips = atoi(strtok(NULL, "-"));
  trips = new Trip[numTrips];
  blocks = readFile(inf, citySize, blockCount, trips, numTrips);
  ct.reset();
  streetSelecter = new StreetSelecter(blocks, blockCount, citySize, numTrips);
  delete [] blocks;
  streetSelecter->select(trips, numTrips, widenedBlocks, numWidened);
    // fills widenedBlocks
  cout << "CPU Time: " << ct.cur_CPUTime() << endl;
  writeSolution(argv[1], widenedBlocks, numWidened);
    // writes widenedBlocks to filename.ans
  sprintf(command,"RunTrips.out %s", argv[1]);
  system(command);  // runs trips of file using original and widened streets
  return 0;
}

class StreetSelecter
{
public:
  StreetSelecter(Block *blocks, int blockCount, int citySize, int numTrips);
  void select(Trip *trips, int numTrips, Trip widenedBlocks[100], int
&numWidened);
    // fills widenedBlocks and sets numWidened to the number of entries
}; // StreetSelecter class
```