CS 5334/4390 Spring 2017
Shirley Moore, Instructor
Lab 2 Due March 10

OpenMP Parallelization of the Floyd-Warshall Algorithm

The Floyd-Warshall (FW) algorithm solves the all-pairs shortest path problem on directed graphs. Assume we have a directed weighted graph G = (V, E, W), where V = {1, …, n} is a set of vertices, $E \subseteq V \times V$ are the edges, and W is a weight or distance function E -> R.
The standard FW algorithm is shown below. The algorithm assumes that A_0 is initialized with

$A_0[i,i] = 0$ for all i,
$A_0[i,j] = W(e_{ij})$ if $e_{ij} \in E$
$A_0[i,j] = \infty$ if $e_{ij} \notin E$

At the conclusion of the algorithm, the $A_n[i,j]$ entry is the length of the shortest path between vertices i and j. FW is used in many real-life applications, such as in bioinformatics for clustering correlated genes, in database systems for optimizing SQL queries, and in data mining.

The Floyd-Warshall (FW) algorithm

```
1: for k = 1 to n do
2:    for j = 1 to n do
3:        for i = 1 to n do
4:            Ak[i, j] = min(Ak-1[i, k] + Ak-1[k, j],Ak-1[i, j])
5:        end for
6:    end for
7: end for
```

A nice feature of the FW algorithm is that it works with negative weights, as long as there are no zero or negative length cycles. Note that there is a dependency between iterations of the outer k loop, but that iterations of the i and j loops are independent. In the implementation, only one copy of A need be kept, with each iteration of the k loop overwriting the previous value. If desired, an auxiliary path array can be kept to allow recovering the shortest paths.

For this lab assignment, you are to implement the algorithm in C and then use OpenMP and vectorization to parallelize it as efficiently as possible. Your program should input n and the rows of E, one row of E per line. Assume that the values in E are single-precision real numbers. Output the result matrix A in the same format. You should also use the OpenMP timing routines to time just the computational part of your program, not including the time for input and output. You can use files for input and output by redirecting stdin and stdout, respectively. In addition to your code, a README file, and a Makefile, you should turn in a report analyzing the parallel speedup and efficiency of your code. Your program will be tested on Stampede. A prize will be given for the program that runs the fastest on one node of Stampede.

Required for graduate students and extra credit for undergraduates: In addition to the version above for outputting the matrix A with the lengths of the shortest paths, provide an option –paths for outputting the shortest paths between each pair of vertices. Do not time this version.