CSC346/646 Parallel Computation Spring 2015

The K-Clique Problem

For our second project, we will develop a parallel program using OpenMP to solve a K-Clique problem. The K-Clique problem is well known to be $\mathcal{NP}$-complete.[1]

Your main program should accept the name of a data file and a positive integer k on the command line. E.g. `% clique datafile 8`.

Definition: Let $G = (V, E)$ be an undirected graph.

- A k-clique of G is a subgraph with k vertices that are completely connected. I.e., for any two vertices v and w in the k-clique, the edge (v, w) is in E.
- The k-clique problem can be stated as follows: Given an undirected graph $G = (V, E)$ and a positive integer k, where $3 \leq k \leq |V|$, does G have a k-clique ? If so, which vertices form the k-clique ?

Data Format:

The sample test data file "graph9.gdat" is a text file. It can be downloaded from *http://menehune.opt.wfu.edu/csc346*. On the first line, there is one integer n, indicating the number of rows and columns in the square adjacency matrix A representing an undirected graph. The following n lines each contain n numbers (either 0 or 1) indicating the entries in matrix A. If $A_{i,j} = 1$ then there is an edge from vertex i to vertex j in the graph. The graph contains no edges from a vertex to itself.

The file "graph9.gdat" represents a graph with 64 vertices.

Your Tasks:

1. Write a parallel program using OpenMP to solve the k-clique problem.
2. Run your program on a sample input file "graph9.gdat" to find a 9-clique. Run with number of threads in the range 1 to 32. Measure your run times and plot a speedup curve.
3. Save your answer in a file named "solution".
4. When complete, make a ".tar" archive of your directory (a.k.a. 'folder') and upload it to your account on telesto.cs.wfu.edu

For the sample problem in the file "graph9.gdat", an 8-clique was found on gottlieb in 34 seconds; a 9-clique required 4 minutes. There is no 10-clique in the sample graph. Verifying that there is no 10-clique in "graph9.gdat" required approximately 25 minutes.

Hint: While developing and debugging your program, try finding smaller cliques, e.g. size 7.

A sample program to generate indices for all subsets size k selected from a set of n objects is shown on the next page.

[1] In practical terms, a solution to an $\mathcal{NP}$-complete problem can be recognized in polynomial time, but in the worst-case requires exponential time to discover.

```
#include <iostream>
#include <cstdlib>
using namespace std ;

// ------------------------------------------------------------
int showz( int z[], int k )
{
   for ( int i = 0 ; i < k ; i++ ) cout << z[i] << " " ;
   cout << endl ;
}

// ------------------------------------------------------------
void rgen( int z[], int p, int n, int k )
{
   int i, i_start, i_end ;

   if ( p == k ) {    // Base case.
     showz(z, k) ;
   }
   else {
      i_end =  n - k + p + 1  ;

      if ( p == 0 ) i_start = 0 ;
      else i_start = z[p-1] + 1 ;

      for ( i = i_start ; i < i_end ; i++ ) {
           z[p] = i ;
           rgen( z, p+1, n, k ) ;
      }
   }
}

// ------------------------ M A I N ------------------------
int main()
{
    int z[ 32 ] ;
    rgen( z, 0, 8, 3 ) ;
}
```

Sample output begins:

```
0 1 2
0 1 3
0 1 4
0 1 5
0 1 6
0 1 7
0 2 3
0 2 4
0 2 5
0 2 6
0 2 7
  .
  .
  .
```