

# Documentation for Maximal Clique Enumeration (in TODS'11)

## Contents

<b>1</b>	<b>Program Logic</b>	<b>2</b>
<b>2</b>	<b>Compilation and Usage</b>	<b>2</b>
2.1	Supporting Operating Systems . . . . .	2
2.2	How to Compile . . . . .	2
2.3	Usage . . . . .	3
<b>3</b>	<b>Data Format</b>	<b>3</b>
3.1	Input Data . . . . .	3
3.2	Output Data . . . . .	4
<b>4</b>	<b>References</b>	<b>4</b>

# 1 Program Logic

**MCE\_TODS.h** This file defines the set of global variables, structures and functions. Those global functions are listed as follows:

- `allocatememory(size_t size)`, which checks whether there is enough memory in temporal memory block.
- `allocatestaticmemory(size_t size)`, which checks whether there is enough memory in static memory block.
- `ReadGraph(Entry &G, int &gnodenum, int nodenum, int memory-bound, FILE inFile)`, which reads input graph from disk and extracts subgraphs
- `ComputeClique(Entry &G, FILE &Wfile, int Umin, int Umax, int nodenum, int maxdeg)`, which computes maximal cliques in subgraphs (B+ graph)

**Runtimecounter.h** This file defines the runtime counter for the program to get the accurate running time.

**MCE\_TODS.cpp** This file defines the main function, which is the entry point of the whole program.

## 2 Compilation and Usage

### 2.1 Supporting Operating Systems

**Linux** In order to install the program, you have to make sure that GCC compiler has been installed in the PC already.

**Windows with gcc or cygwin** Make sure that GCC compiler has been installed in the PC already.

This program has been tested in CentOS 5.4, Ubuntu 10.04. And it should work in other Linux distribution and Cygwin (Not guaranteed ).

### 2.2 How to Compile

1. Copy all source files into one folder
2. Use "g++ MCE\_TODS.cpp [optimization level] -o [program name]". [optimization level] is the optimization level specified by users, could be -O3 or -O0 for example. We tested our program using -O3 setting. [program name] is the user specified name for the executable. We also provide an example of makefile.

## 2.3 Usage

Our Program accept 5 arguments. They should follow required order. If *program* is the name of our program. The usage should be like:

program [graph\_file\_name] [memory\_bound(MB)] [Bsize] [average\_degree] [output\_or\_not]

- [graph\_file\_name]: (string) Input graph file name
- [memory\_bound\_MB]: (int) Memory bound for H graph, in MB
- [Bsize]: (int) the upper bound for number of B-vertices.
- [average\_degree]: (int) average degree of graph
- [output\_or\_not]: (int)
  - 0 - NOT output cliques, just count the total number
  - 1 - output all maximal cliques

## 3 Data Format

### 3.1 Input Data

Assume  $u$  denotes a node and  $d$  denotes its degree, the format of graph data is : u,d: $v_1:v_2:\dots:v_n$ . **And also  $v_1, v_2, \dots, v_n$  are in an order.**

As an example, a graph with 10 nodes is represented as follows, where node 0 has 1, 2, 3, 6, 7 and 8 as neighbors.

```
10
0,6:1:2:3:6:7:8
1,6:0:2:3:4:6:7
2,7:0:1:3:4:6:7:8
3,5:0:1:2:4:9
4,5:1:2:3:5:8
5,2:4:8
6,4:0:1:2:7
7,4:0:1:2:6
8,4:0:2:4:5
9,1:3
```

Note:

1. Pay attention to the encoding format of graphdata file, especially in windows OS. We use MS-DOS kinds .txt file, i.e., each line is ended with '\n' and before '\n' there is an additional character with ascii code 10.
2. Pay attention to the node id range. The node id of each node must be in the range of [0, nodenum-1].
3. **Pay attention to the ids of neighborhood nodes. The neighbors of each node are ordered by their ids in ascendant order.**

### 3.2 Output Data

Output of this program is in plain text. Each line consists a maximal clique by listing all the node *ids*.

## 4 References

1. J. Cheng, Y. Ke, A. W.-C. Fu, J. X. Yu, and L. Zhu. Finding maximal cliques in massive networks. *To appear in ACM TODS, 2011*