

CS 4000  
**Homework # 3: Minimum Weighted Vertex Cover and  
C++ Threads**

due Tuesday Feb. 23rd, 2016  
(25 pts.)

Given a graph  $G = (V, E)$  and a weight assigned to each vertex (described by a function  $w : V \rightarrow \mathbb{N}$ ), we would like to find the MINIMUM WEIGHTED VERTEX COVER for the graph. (A vertex cover in a graph  $G$  is a set of vertices  $V' \subseteq V$  such that for every edge  $\{u, v\} \in E$ , either  $u \in V'$  or  $v \in V'$ .) As we did in Lecture #8, we will search through all possible subsets for the best one. A reference implementation `MWVC.cc` that solves this problem is provided along with this homework.

For this assignment, you will modify the code in `MWVC.cc` to run in parallel using the thread features in the C++ 2011 (and above) standard. Your modified code must do the following.

1. Produce the correct result.
2. Have no race conditions or other concurrency errors.
3. Query the `hardware_concurrency` function in the `thread` class to determine how many threads to create.
4. Create multiple threads to solve the problem. Your code should create as many threads as possible (consult the answer you received from #3.) Furthermore, each thread created should return a value using the `future` class in C++ 11.
5. Be at least 75% efficient when compared with the reference implementation.
6. Be documented appropriately. Please clearly indicate what code was added/modified to the reference implementation.

Submit your modified code in a single file `MWVC_with_threads.cc`.