

**COMP 2355**  
**Winter 2012**  
**Homework #4**  
**Due at midnight, March 12, 2012**

For this assignment you need to modify the code from Homework #3. An official solution to Homework #3 will be posted on the course web page after the deadline for Homework #3. In this homework you will modify your previous solution to run in parallel. *Note that you can use the simpler HW #3 code that does not use paging for storing data on disk.*

1. As in Lab 13, your main BFS function should distribute states to threads to be expanded, but now you must write the updated depth/status back to memory. When doing this, the following precautions should be taken:
  - a. Only one thread should be writing to any portion of the depth/status bit arrays at one time. Locks should be used to prevent more than one thread from writing to these data structures concurrently. You can use one lock for the whole array, but it will be more efficient to use multiple locks for different regions of the array.
  - b. Threads should cache their writes locally, and only write to the main data when enough states have been cached for writing.
  - c. When data is sent to the threads, certain data values should be reserved for sending messages to threads. For instance, a message should be used to indicate that the current depth or the current bucket is finished. (This could, for instance, cause the threads to flush any cached data.)
    - i. You may want separate queues for each thread to avoid one thread reading the same message twice. If you do this, though, you might have to load-balance the work you send to threads to get the best performance.