This Fibonacci Heap implementation was developed in MATLAB for general use but with the specific aim of later integration with the Dijkstra's algorithm implementation that is used by Matlog[1].

**Instructions** To use the MATLAB Fibonacci Heap, one first must place the accompanying MATLAB script files (`cFibHeap.m` and `cFibHeapNode.m`) into the following subdirectories, where `dirInPath` is any directory in the MATLAB path:

- `dirInPath/@cFibHeap/cFibHeap.m`
- `dirInPath/@cFibHeapNode/cFibHeapNode.m`

To create a heap named `myHeap`, one should execute the following MATLAB command: `myHeap=cFibHeap`. Keys (values) can be inserted into the heap using the command `myHeap.insert(num)`, where the key to be inserted is `num`. The command `myHeap.findMin` returns the minimum key, and the command `myHeap.extractMin` returns and removes the minimum key. The size of the heap (total number of nodes) can be determined by the command `myHeap.n`. The operations `insert`, `findMin`, and `extractMin` are the only operations presently supported in this beta version of the MATLAB Fibonacci Heap.

**Future Release** Further analysis is needed to optimize the MATLAB code. The implementation must be modified in order to be integrated into Matlog's Dijkstra's implementation; in particular, the code must be modified to allow the storage not only of keys, but also of corresponding pointers to paths being considered during the operation of Dijkstra's algorithm. The following standard heap operations will be supported in a future release: Merge-Heaps, Increase-Key, and Decrease-Key. For more information on Fibonacci Heaps and standard heap operations, the reader is referred to a standard computer science textbook[2] on algorithms.

[1] http://www.ise.ncsu.edu/kay/matlog/

[2] Cormen, Leiserson, Rivest, and Stein. *Introduction to Algorithms.* 2nd Ed. McGraw-Hill (2007).