**Chapter 1: Intro**

* What are graphs?
* What & Why betweenness centrality?
* Why are algorithms different?
* Why test them?
* What I did
* Past work

**Chapter 2: Preparation**

1. Chosen Algorithms
2. Requirements Analysis
   1. Deployment: my tests + researchers
   2. How will it accomplish it’s objective (measurability)
      1. Datasets
   3. Uses (tests)
   4. Effectiveness (comparable to best known implementation, but not multithreaded)
   5. Environment (high performance server)
   6. Components
   7. Chart of components? (algorithms, harness, tests, datasets)

* The Algorithms
* Demonstrate professional approach
* *Requirements Analysis*
* Tools I could use (discuss language)
  + Multicore
  + Instrumentation
  + Server
* Engineering practices
  + Verification
    - The test package
    - JGraphT
    - Relative performance
    - Graph checking (disconnected, multigraph?)
    - All in theory
  + Source control
  + Testing
  + What was the best practice and how did I compare?

**Chapter 3: Implementation**

* What I actually wrote (graph representation, algorithms,instrumentation)
* How source control was helpful (my branches, etc)
* Graph representation struggle
* Heaps
* How testing was helpful
* Server
* More algorithms
* Functionality and Limitations (graph size, per-node instrumentation)
* *Repository Overview*
  + What was written from scratch vs what wasn’t

**Chapter 4: Evaluation**

**Chapter 5: Conclusions**