Detecting Concurrency Errors with JPF
Questions about JPF?
Configurations

- Default JPF: exhaustive search (DFS)
  - Threads scheduled in the order given by their IDs

- Random search order
  \[+cg\text{.randomize}\_choices=VAR\_SEED\]

- Preemption bounding
  \[+vm\text{.scheduler}\_sync\_class=gov\text{.nasa}\text{.jpf}\text{.vm}\text{.schedule}\_ContextBoundingSyncPolicy\]
  \[+vm\text{.scheduler}\_sharedness\_class=gov\text{.nasa}\text{.jpf}\text{.vm}\text{.schedule}\_ContextBoundingSharednessPolicy\]
  \[+contextbound\_max\_number\_of\_preemptions=<N>\]
Configurations

- **Breadth first search (BFS)**
  
  ```
  +search.class=gov.nasa.jpf.search.
  heuristic.BFSHeuristic
  +search.heuristic.queue_limit=-1
  ```

- **Maximize thread preemption**
  
  ```
  +search.class=gov.nasa.jpf.search.
  heuristic.Interleaving
  ```

- **Minimize preemption**
  
  ```
  +search.class=gov.nasa.jpf.search.
  heuristic.MinimizePreemption
  ```

- **Maximize blocked threads**
  
  ```
  +search.class=gov.nasa.jpf.search.
  heuristic.MostBlocked
  ```
What to do now

- Finish remaining tasks from the last seminar
  - Writing reasonable environment for `LinkedList` and `Semaphore` (try different workloads)

- Play with different configurations aiming at efficient detection of concurrency errors
  - Use additional benchmark programs (examples)