Final Exam Topics

1. **Data Parallel**
   - MasPar Example
   - Parallel Prefix and Parallel Linked List Length
2. **Semaphores**
   - Abstraction with two services P (wait) and V (signal)
   - Critical section problem and semaphores
   - Java synchronized and semaphores
   - Barriers and semaphores
   - Producer / Consumer Problem; Dining Philosophers Problem; Reader/Writer Problems
3. **Monitors**
   - monitors and conds
   - wait(cv), wait(cv, rank), signal(cv), signal_all(cv), empty(cv), minrank(cv)
     - signal and wait versus signal and continue
     - queues, priority queues, BPOTs, heaps and analysis
     - bitonic lists
   - signal and wait versus signal and continue
   - semaphores implemented via monitors
   - monitor examples
     - semaphores, bounded buffers, readers/writers, shortest-job-next, sleeping barber
     - CSCAN/SCAN disk scheduler (bitonic lists)
4. **Java Support for Monitors**
   - Synchronize : specifies critical section using an object as lock
     - can do at granularity of method
     - can do at granularity of a block
   - Java synchronized, wait/notify/notify_all
   - Locks are reentrant
   - Locks can be temporarily given up : wait and notify
5. **Single lane bridge problem using semaphores and monitors**
6. **Message Passing**
   - channels: send (non-blocking); receive (blocking); empty
   - simple channel examples: char-to-line; sorting network
   - client server examples
   - MPI
7. **Parallelizing Graph Algorithms**
   - All shortest paths (Floyd's)
     - Cannot parallelize pivots
     - Barriers for various approaches
   - Minimum spanning tree (Prim's Algorithm)
     - alternate data structures for adjacency (N^2 versus E lgN)
   - Block Striped Partitioning
   - Analysis of Prim's using p processors
     - computation cost N^2/p
     - communication cost
       - Hypercube O(N lg p); use p = N/lg N
       - Mesh $O(N p^{½})$ use $p = N^{2/3}$

8. **Distributed Computing Paradigms**
    - Channels (all the ways down to UDP and TCP/IP)
    - Distributed Objects
    - Mediated -- Spaces and Message Queues
9. **UDP, Multicast UDP, TCP/IP**
    - Concepts, comparisons
10. **Concurrent Objects**
    - Synchronous vs asynchronous
    - Garbage collection when distributed
11. **RMI**
    - use of interface
    - serialization
    - handles to remote objects (stubs)
    - garbage collection
    - Bid.com via RMI
12. **Tuple Space**
    - read (rd), take (in), write (out), eval
    - readIfExists (rdp), takeIfExists (inp)
    - leases on tuples
    - Bid.com as a tuplespace
13. **Atomicity and Transactions**
    - Commit/Abort; roll forward/roll back
14. **Object Request Broker (ORB)**
    - Discovery, Join, Lookup
    - Discovery process
    - Packet storms on restart
15. **Oblivious Comparison Exchange Sorts**
    - Proof of correctness for 0-1 data implies correct for all
    - Correctness of Even-Odd Transposition Sort
16. **Shear Sort and its Analysis**
    - Shear Sort and RevSort
    - Order, Cost, Work, Cost Efficiency, Work Efficiency.
17. **Revsort (a kin of ShearSort)**
    - Extend shear notion to the technique used in Revsort.
      Note this is not a snake sort like shear.
    - Revsort is not a sort. It just gets close (within 8 rows of being right.)
    - Revsort gets there fast. It cuts number of dirty rows, not in halves, but to square root of current number of dirty ones.
18. **Bitonic Sort**
    - Mapping to hypercube
19. **Virtualizing Algorithms**
    - Brent Scheduling, but not just for binary tree reduction
20. **Accelerated Cascading**
    - lg lg N max
    - tradeoff points
21. **PCN (Program Composition Notation)**
    - mutable vs definitional
    - intentional non-determinism

22. **CSP**
    - guarded communication
23. **Parallel Constraint logic programming**
    - generators and consumers
24. **Program Flow Analysis**
    - basic concepts (e.g., basic blocks, intra and inter procedural, aliasing)
    - flow graph
        - DFS numbering
        - domination
        - du and ud chaining
        - forward/backward and may/must
    - Parallelizing code
        - scalar dependence (true, anti and output)
        - Diophantine analysis
            - GCD Test
        - Vectorizing loops
25. **Scheduling Algorithms**
    - General Problem -- times and partial order
    - Timing (Gantt) Charts List Schedule
      Sorting when no partial order
    - Anomalies
    - UET trees and anti-trees (breadth first order)
    - UET graphs and 2 processors
    - NP Completeness
26. **Message Ordering**
    - Receive, Priority, Time, Causal, CATOCS