CSE 109 – Final Review

1. Midsize Program Building
   a. What is a Makefile and why do we use them
   b. What is the standard way to organize a project
   c. What is the purpose of a README
   d. What goes in the lib directory?
   e. What goes in the src directory?
   f. What goes in the include directory?
   g. Why do we break a program into multiple files?
2. Lifecycle of an executable
   a. Source
   b. Lex
   c. Parse
   d. Compile
   e. Assemble
   f. Link
   g. Load
3. Datastructures
   a. Big O notation
   b. Hash Sets
      i. What is the advantage of a HashSet
      ii. What is the disadvantage of a HashSet
      iii. What are some applications of a HashSet
      iv. What design choices can you make that affect the performance of a HashSet
      v. What is a hash algorithm?
      vi. What is a prehadh algorithm?
   c. B-Trees
      i. What is a b-tree?
      ii. When would you use a b-tree?
      iii. What is the advantage of a b-tree over a hash table/set?
4. Debugging
   a. What is a debugger?
   b. What is gdb and why is it useful?
   c. What is a breakpoint and why is it useful?
   d. What is a stack trace and why is it useful?
   e. Why not just printf your way through life?
   f. What role does program state play in debugging? Why are we interested in looking at program state? What about the design of C/C++ makes inspecting state difficult?
5. Networking
   a. What is a network?
   b. What is a protocol and why do we need them?
   c. How do we use protocols in everyday life?
   d. Why can’t we just send data to a server and expect the server to understand?
   e. What is compression/decompression and why do we do it?

f. What resources do we have to be concerned with in a networking context
g. What additional dangers are there w.r.t. transmitting and receiving data?
h. How do we send a C++ object or struct over a network?
i. What is serialization / deserialization