1. Suppose you have a formula that is necessarily true which means that it evaluates to true for any assignment of the variables. (For example, \((A \lor \overline{A} \lor C) \land (B \lor \overline{B} \lor C)\) is necessarily true. What will be the run time for GSAT on such instances?

2. GSAT doesn’t bother to store previously tried random truth assignments, which means that it could repeat the same random assignment. Why do you think it is implemented that way?

3. It may seem at first that an algorithm which returns “yes” or “I don’t know” is not very useful. Describe a scenario (possibly by choosing an NP-hard problem), and some assumptions, where such an algorithm could be useful.

4. What is the difficulty collecting data for large \(N\) for graphs of the type shown by Monasson et al.?

5. Why not look at the ratio of clauses to variables (or ratio of 2-clauses to 3-clauses) and simply predict the solution using the graphs in Monasson et al. (at least for the extreme cases) rather than going through the trouble of solving SAT instances?

6. What do Monasson et al. mean by a backbone?

7. Suggest an additional question for discussion.