In this project you will get some experience with uncertain inference by implementing some of the algorithms for it from the textbook and evaluating your results. We will focus on Bayesian networks, since they are popular, well-understood, and well-explained in the textbook. Recall that the inference problem for a Bayesian network is defined as follows:

Given a Bayesian network over the random variables \( \{X\} \cup E \cup Y \), such that

- Query variable is \( X \),
- Evidence variables are \( E \),
- Values for the evidence variables are \( e \),
- Unobserved (or hidden) variables are \( Y \),

Calculate \( P(X | e) \), that is, the conditional distribution of the query variable given the evidence. That is, the probability for each possible value of the query variable, given the evidence.

In general, we have that:

\[
P(X | e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y) \tag{AIMA Eq. 13.9}
\]

And for a Bayesian network you can factor that full joint distribution into a product of the conditional probabilities stored at the nodes of the network:

\[
P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | \text{parents}(X_i)) \tag{AIMA Eq. 14.2}
\]

These equations are the basis for the various inference algorithms for Bayesian networks that we’ve seen in class and in the textbook.
Your assignment is as follows:

1. You need to implement an **exact inference** algorithm for inference in Bayesian networks. I recommend the “inference by enumeration” algorithm described in AIMA Section 14.4. Pseudo-code for the algorithm is given in Figure 14.9. Section 14.4.2 suggests some speedups for you to consider.

2. You need to implement an **approximate inference** algorithm for inference in Bayesian networks. The algorithms described in the textbook and in class are: (1) rejection sampling, (2) likelihood weighting, and (3) Gibbs sampling. We have seen that the first two can be quite inefficient. Gibbs sampling is not that hard, except for the bit explained at the bottom of AIMA page 538.

Whichever algorithms you choose, explain your choice, explain your design and implementation, and evaluate the results. You might implement more than one algorithm in each category and compare them. You should compare the approximate algorithm(s) to the exact algorithm(s). You should evaluate and describe the performance of the approximate algorithms as the number of samples increases.

As always, your grade will be 50% for your code and 50% for your writeup. You know what I expect in a writeup by now some I won’t repeat that here.

I suggest you start by working on the Burglary/Earthquake Alarm problem (AIMA Figure 14.2) since it is well-described in the book and we did it in class. I will try to make available further problems for your use, and perhaps some code to help you get them into your inference engines. You can also find (and share) additional examples of Bayesian network problems, but of course please don’t go near the code you can also easily find on the web.

Submit a zip or other archive of your work using Blackboard before the deadline. Include your writeup as a PDF in the archive. As before, please name your archive starting with your last name (e.g., ferguson-project-03.zip).