

## Part1 function

### **After\_evidence(evidence, factors)**

This function returns the factors dictionary after passing all evidence.

### **Find\_topo\_order(proposal\_factors)**

This function returns the sample order of the directed graph. The core query is **nodes = nx.topological\_sort(G)**.

### **Cal\_joint\_dist(proposal\_factors)**

This function returns the joint distribution.

### **Get\_prob(samples, factors)**

With sample dictionary and factor, we can get the exact probability.

### **\_sample\_step(nodes, proposal\_factor):**

This function returns sample dictionary.

### **\_get\_conditional\_probability(target\_factors, proposal\_factors, evidence, num\_iterations)**

This function will 1. Find the sample order, find the r dictionary, record sample result and frequency, and return the conditional probability.

## Part2 function

### **\_sample\_step(nodes, factors, in\_sample)**

This function performs Gibbs sampling.

### **\_get\_conditional\_probility(nodes, edges, factors, evidence, initial\_samples, num\_iterations, num\_burn\_in)**

This function first update the evidence information in the factors list. Then do (burn\_in+num\_iterations) sampling. Then count the frequency of the whole iteration. And calculate the output factor.