Algorithm Researcher Programming Test

Introduction

An Ising spin model can be thought of as a graph with nodes and edges, in which each node has a spin which can be positive (+1) or negative (-1). The spin of node v is denoted s_v. The nodes have weights given by a vector h and the edges have weights given by an upper triangular matrix J; weights can be positive, negative, or zero. The spin configuration, i.e., whether each node's spin is positive or negative, is known as the *state*. The energy of the system depends on the state and the weights given by h and J. The equation for the system energy given by state $S = (s_1, \dots, s_n)$ is:

$$E(S) = \sum_{1 \le i \le n} h_i s_i + \sum_{1 \le i < j \le n} J_{ij} s_i s_j$$

We are generally interested in finding a *ground state*, i.e., a state that minimizes the energy of the system. Put another way, a state S_g is a ground state if and only if $S_g = \arg\min_S E(S)$. The energy of a ground state is known as the ground state energy, and is equal to $\min_S E(S)$. Thus the problem of finding a ground state is a minimization problem.

The algorithm

This problem on general graphs is NP-complete, but on trees it is solvable in polynomial time using dynamic programming. Make the tree rooted by assigning any node as the root. Let $S(v, \sigma)$ denote the minimum energy configuration of spins for nodes in the subtree rooted at node v, given that the parent of node v has spin σ. It is also convenient to consider $E(v, \sigma)$, which is the energy of this sub-configuration. This is the sum of energy contributions of all nodes and edges in the subtree rooted at node v, along with the edge between node v and its parent. When v is a leaf node, it should be straightforward to determine $S(v, \sigma)$ and $E(v, \sigma)$. When v is not a leaf node, but we know, for every child u of v, $S(u, \sigma)$ and $E(u, \sigma)$, it should again be straightforward to calculate $S(v, \sigma)$ and $E(v, \sigma)$.

Write a program that computes a ground state (as a vector of spins) and the ground state energy for an Ising spin model on a tree.

The program can be written in Python, C/C++, or a combination. The algorithm should require no more than $O(n \log n)$ time and space. In particular, since J is an extremely sparse matrix, it should not be stored as a full matrix. Apart from the asymptotic time and space requirements, optimizing for speed is not necessary. We recommend using pure Python for simplicity.

Feel free to make use of any online resources, but please avoid seeking personal help, e.g., asking friends or posting on forums.

Input format

Each input will be contained in a separate file containing text lines defined as follows.

- A comment line starts with a lower case c. Comment lines are for annotating the files and can be ignored by your program. Comments may appear anywhere in the file.
- The problem line starts with a lower case p. This line must appear before any data lines. It contains three fields: the test name, the number of spins (i.e., n), and the number of weights (non-zero elements of h and J). Fields are separated by at least one space. The test name is an ascii string identifying the problem instance. The number of weights is equal to the number of data lines in the file.
- After the problem line there are some number of data lines. There are three fields: u, v, weight. If $u \neq v$ the (J) weight is assigned to the edge between nodes u and v. If u = v, the (h) weight is assigned to that node. You may assume the weights are integers.
- Here is an example file:

```
c This is an extra-hard problem

p test01 4 6

0 1 1

1 2 1

1 3 1

0 0 -1

1 1 -1

2 2 -1
```

Output format

The output should consist of two lines. The first should contain the ground state energy. The second should contain a ground state as a series of + and -. Output for the above problem could be:

```
-4

+-++
```

Submission

Please email us your source code. If you've generated other relevant material, such as a Makefile or test cases, please include it.