Problem G
Gambling Guide

Submits: 41
Accepted: at least 16

First solved by: UW1
University of Warsaw
(Dębowski, Radecki, Sommer)
01:30:08

Author: Gustav Matula

Problem:
You're located at a node in an undirected graph.

In each step a neighboring node is chosen at random, and you can either move there or stay where you are.

Find the expected number of steps to get from node 1 to node N, if you used an optimal strategy.

Assume we knew $f(x)$ - the expected number of steps to get from node x to node N.

The optimal strategy to use at each node x is then an obvious one: when offered to move to a neighbour y, move if $f(y) < f(x)$, and stay otherwise.

But we don't know $f(x)$, except for $f(N) = 0$.

Let S be a set of nodes for which we know the value of $f(x)$. Starting from $S = \{N\}$, we'll keep adding nodes one by one in the order of increasing values $f(x)$.

To find the next node to add, we consider nodes outside of S, but neighbouring some node in S. Compute the f'(x) for each such node following the strategy as if that node is the next to add (i.e. move to nodes in S, or stay otherwise).

$$f'(x) = 1 + \sum_{\text{neighbour } y \in S} \frac{f(y)}{degree(x)} + \sum_{\text{neighbour } y \notin S} \frac{f'(x)}{degree(x)}$$

$$f'(x) = \frac{degree(x) + \sum_{\text{neighbour } y \in S} f(y)}{degree(x) - \sum_{\text{neighbour } y \notin S} 1}$$

The node with minimal f'(x) is the next to add.
We set f(x) = f'(x) and add x to S.

We end up with an algorithm very similar to Dijkstra's single source shortest path algorithm, and we can implement it efficiently using the same techniques.

Complexity: $O((N + M) \log N)$ using the classic implementation with a binary heap (or STL set).