Date No.

2. Proving in NP.

Assume that the certificate given follows the rules of PLUGIT, which it is a valid-instance.

The certificate takes the form of a list l, where the elements in l are the powered sockets.

Now let U be the set of all the sockets. The verification algorithm loops through the powered socket list l, for each powered socket find all the other sockets that can be powered via this current socket, either by direct wire connection or powering of nearby broadcast nodes. Let S call this set of derived powered socket S. Then, remove S from U. When the algorithm terminates, simply check if U is empty, i.e. no unpowered sockets left, and return yes and no accordingly.

It is clear that the looping of sockets take poly time, and the finding of removing of sockets take $O(|U|)$ worst. Since encoding issues and $\log n$ factors of inputs are not a concern of PLUGIT, the verification algorithm runs in polynomial time. The certificate is in poly size too, since it is just a list of unpowered sockets.

With question 1 proving PLUGIT is NP-hard and question 2 giving a polynomial time verification algorithm, and polynomial sized certificate, it is shown that PLUGIT is NP-complete.