Problem 1. Give a deterministic algorithm for the language

\[ LP = \{ \langle G, s, t, k \rangle \mid \text{There is a path in } G \text{ from } s \text{ to } t \text{ of length } k \}, \]

where \( G = (V, E) \) is an unweighted graph, \( s \) and \( t \) belong to \( V \), and \( k \in \mathbb{N} \). Here, the length of a path is the number of edges you cross over from its beginning to its end.

Solution 1.

If \( k = 0 \), accept if and only if \( s = t \).
If \( k = 1 \), accept if and only if \( (s, t) \in E \).
Otherwise, for each \( p \in V^{k+1} \), check if \( p \) is a path from \( s \) to \( t \). If so, accept.
Reject.

Problem 2. Nondeterministic Turing machines are best used for sufficiently complex languages that we can benefit from exploring multiple paths. A consequence of this is that implementing a nondeterministic Turing machine at its lowest level via the formal definition is somewhat of a nightmare. As such, we usually only give pseudocode algorithms for them.

Give a nondeterministic Turing machine (you need not draw one) which decides \( LP \) at least exponentially faster than the deterministic version.

Solution 2.

If \( k = 0 \), accept if and only if \( s = t \).
If \( k = 1 \), accept if and only if \( (s, t) \in E \).
Pick a \( p \in V^{k+1} \) nondeterministically.
Check if \( p \) is a path from \( s \) to \( t \). Accept if it is and reject otherwise.

Problem 3. Explain in your own words why it is hard to flip the output of a nondeterministic Turing machine.

Solution 3. Because an NTM rejects only if every branch of its computation rejects, it is easy to flip a reject to an accept. However, an NTM accepts if any branch of its computation accepts. This means there can be several branches of computation which reject. Any one of these would cause the flipped output to accept since the flipped output should reject only if every branch of computation accepts. This means we have to explore every branch of computation to determine if we should flip an accept to a reject or if it should remain an accept.