Theorem 1 (3.10.2). Algorithm NAIVE-MINIMIZE terminates and correctly returns an array

\[
U({p, q}) = \begin{cases} 
1, & \text{if } p \not\equiv q; \\
0, & \text{if } p \equiv q.
\end{cases}
\]

Furthermore, the pair \{p, q\} is marked at the n’th iteration of the while loop if and only if the shortest string distinguishing p from q is of length n.

Proof. Clearly the algorithm terminates, since there are a finite number of pairs and so eventually we make it through the while loop starting on line 3 without marking any new pairs.

We will prove, by induction on \(n\), that (*) the pair \{p, q\} is marked by the algorithm at the n’th iteration iff \(p \not\equiv q\) and the shortest string distinguishing \(p\) from \(q\) is of length \(n\).

The base case is \(n = 0\). Then \{p, q\} is marked after doing 0 iterations iff \(p \in F\) and \(q \in Q - F\) (or vice versa), which occurs iff \(\epsilon\) distinguishes \(p\) from \(q\), which occurs iff the shortest string distinguishing \(p\) from \(q\) is of length 0.

For the induction step, suppose the assertion (*) holds for all \(n' < n\). We prove it for \(n\).
Suppose \( \{p, q\} \) is marked at iteration \( n \) in step 8. But marking at step 8 occurs only if there exists a letter \( a \) such that \( \{p', q'\} \) is already marked, with \( p' = \delta(p, a) \) and \( q' = \delta(q, a) \). In fact, this marking of \( \{p', q'\} \) must have occurred at iteration \( n - 1 \); otherwise we would have considered the pair \( \{p, q\} \), and then marked it, at some iteration \( < n \). But if \( \{p', q'\} \) was marked at iteration \( n - 1 \), then by induction \( p' \neq q' \), and the shortest string distinguishing \( p' \) from \( q' \) is some \( t \) with \( |t| = n - 1 \). Then the string \( at \) distinguishes \( p \) from \( q \), and \( |at| = n \). From step 6 we know the pair \( \{p, q\} \) was not marked at any previous iteration, so by induction there is no string of length \( < n \) distinguishing \( p \) from \( q \). So \( at \) is actually the shortest such string.

\[\equiv: \text{For the converse, suppose } p \neq q, \text{ and } x \text{ is a shortest string distinguishing } p \text{ from } q, \text{ and } n = |x|. \] We need to see that the pair \( \{p, q\} \) gets marked at iteration \( n \). Write \( x = ay \) with \( |y| = n - 1 \) and \( a \in \Sigma \), and furthermore let \( p' = \delta(p, a) \) and \( q' = \delta(q, a) \). Now it cannot be that \( p' = q' \), because if so, then \( \delta(p, x) = \delta(p, ay) = \delta(p', y) = \delta(q', y) = \delta(q, ay) = \delta(q, x) \), a contradiction. So \( y \) distinguishes \( p' \) from \( q' \). Furthermore, \( y \) is a shortest such string; if there were a shorter one, say \( y' \), then \( ay' \) would be a string distinguishing \( p \) from \( q \) that is shorter than \( x \), a contradiction. So by induction the pair \( \{p', q'\} \) gets marked at iteration \( n - 1 \). Then the flag \texttt{done} gets set to false at iteration 8, ensuring that one more iteration takes place, where the pair \( \{p, q\} \) gets marked.

\[\square\]