EECS 336 Problem 2.1

Algorithm 1 Myth Booster

Require: List $W[i] = w_i (i = 1, 2, ...n)$ and List $T[j] = t_j (j = 1, 2, ...m)$
Ensure: whether or not

1: **function** MYTHBOOSTER(W, T, n, m)
2:     $p \leftarrow 1$, $q \leftarrow 1$
3:     **while** $q \leq m$ **do**
4:         **if** $W[p] = T[q]$ **then**
5:             $q \leftarrow q + 1$
6:         **end if**
7:         $p \leftarrow p + 1$
8:         **if** $q > m$ **then**
9:             Break
10:         **end if**
11:         **if** $p > n$ **then return** False
12:         **end if**
13:     **end while**
14:     **return** True
15: **end function**

Runtime:

In the while loop, p is incremented by one in each iteration starting from 1. The while loop will be terminated if $p > n$. Therefore, the while loop is executed at most n times. Also, each line in the while loop takes constant runtime. So the algorithm is $O(n)$.

Correctness:

Claim 1: if the algorithm returns False, then T is not a secret message in W.

Proof by contradiction:

Suppose the algorithm returns False but T is a secret message in W. This implies that at least one component in T, which actually has a matching component in W, is found to be unmatched by the algorithm. Let's denote t_k as the first such component in T. So any component in T before t_k (e.g. $t_1, t_2, ..., t_{k-1}$) has been correctly matched with components in W. Let w_{l-1} be the matching word in W of t_{k-1}. Then the supposition implies that, t_k is found unmatched while there is actually a matching word $w_x (l \leq x \leq n)$ in $(w_l, w_{l+1}, ..., w_n)$.

After matching t_{k-1} and w_{l-1}, according to line 5&7 in the algorithm, q was set to k and p was set to l. According to line 5, the algorithm will look at the next word in T only if a matching part of t_k is found. As we assumed that t_k is unmatched, there is always $q = k$ after this point. (*)

During the iteration of while loop, p is incremented by 1. As the algorithm returns False, the while loop terminates as soon as $p > n$ (line 11, e.g. $p = n+1$). Note that $l \leq x \leq n$, so line 5 should be executed when p = x. Then q should be increased by 1 (e.g. q=k+1). This contradicts with (*).

Claim 2: if the algorithm returns True, then T is a secret message in W.

Proof by contradiction:

Assume the algorithm returns True, but the result should be False. That means, in fact, following the words' sequence, at least one component t_k in T cannot find a matching component in W. However, the algorithm incorrectly either (a) matches t_k with certain component w_l in W and then started to look at t_{k+1}, or (b) upon completion of checking t_{k-1}, returns True and ignore the unscanned component t_k to t_m after W has been completely scanned.

Case (a), in which the algorithm matches t_k and w_l and starts to look at component t_{k+1}, contradicts with lines 4-5 in the algorithm. In other words, the algorithm could only look at the next target word after finding a matching pair of words (each from T and W).

Case (b), in which upon completion of checking t_{k-1}, the algorithm returns True and ignores the unscanned components in T after W is fully scanned. As the algorithm returns True, it must reach line 14, indicating that $q > m$. It contradicts with criteria of jumping out of the while loop (line 3 & 8) in the algorithm.

Since (a) and (b) both contradicts the algorithm, the statement is proved.