Let the sequence $S$ consist of $s_1, \ldots, s_n$ and the sequence $S'$ consist of $s'_1, \ldots, s'_n$. We give a greedy algorithm that finds the first event in $S$ that is the same as $s'_1$, matches these two events, then finds the first event after this that is the same as $s'_2$, and so on. We will use $k_1, k_2, \ldots$ to denote the match have we found so far, $i$ to denote the current position in $S$, and $j$ the current position in $S'$.

Initially $i = j = 1$
While $i \leq n$ and $j \leq m$
    If $s_i$ is the same as $s'_j$, then
    let $k_j = i$
    let $i = i + 1$ and $j = j + 1$
    otherwise let $i = i + 1$
EndWhile
If $j = m + 1$ return the subsequence found: $k_1, \ldots, k_m$
Else return that "$S'$ is not a subsequence of $S$"

The running time is $O(n)$: one iteration through the while look takes $O(1)$ time, and each iteration increments $i$, so there can be at most $n$ iterations.

It is also clear that the algorithm finds a correct match if it finds anything. It is harder to show that if the algorithm fails to find a match, then no match exists. Assume that $S'$ is the same as the subsequence $s_{i_1}, \ldots, s_{i_m}$ of $S$. We prove by induction that the algorithm will succeed in finding a match and will have $k_j \leq l_j$ for all $j = 1, \ldots, m$. This is analogous to the proof in class that the greedy algorithm finds the optimal solution for the interval scheduling problem: we prove that the greedy algorithm is always ahead.

- For each $j = 1, \ldots, m$ the algorithm finds a match $k_j$ and has $k_j \leq l_j$.

Proof. The proof is by induction on $j$. First consider $j = 1$. The algorithm lets $k_1$ be the first event that is the same as $s'_1$, so we must have that $k_1 \leq l_1$.

Now consider a case when $j > 1$. Assume that $j - 1 < m$ and assume by the induction hypothesis that the algorithm found the match $k_{j-1}$ and has $k_{j-1} \leq l_{j-1}$. The algorithm lets $k_j$ be the first event after $k_{j-1}$ that is the same as $s'_j$ if such an event exists. We know that $l_j$ is such an event and $l_j > l_{j-1} \geq k_{j-1}$. So $s_{l_j} = s'_j$, and $l_j > k_{j-1}$. The algorithm finds the first such index, so we get that $k_j \leq l_j$. ■

\footnote{ex876.936.4}