Solution
The Algorithm:

```
initialize an index starting at the beginning of word w
initialize an index starting at the beginning of pattern p

begin by scanning w for the first character in p

if that letter is found,
         move w's index to the position after that character
         move p's index to the next letter

repeat until either we reach the end of w,
      or until there are no more characters in p, whichever comes first

if all characters in p have been found in w, return true
```

Proof:
We will now prove that this algorithm is optimal by considering some more optimal solution O, and showing that our algorithm manages to always stay ahead of solution O

First consider the base case where pattern p is of length 1. In this case our algorithm scans word w until it finds character p, and when it does, returns. Because it steps character by character through p, it finds the smallest possible substring that contains p. If there was a smaller substring that matched p, our algorithm would have found it and terminated before looking at any more of the string. Therefore, our greedy algorithm has found a substring as short, or shorter than the optimal algorithm O
Now let consider length of $p > 1$. We will assume the induction hypothesis is true for $p - 1$ and prove it for p. In order to fall behind the optimal algorithm, it would have to examine a substring longer than the substring O examines to match p. However this cannot happen, because, as we have shown in the base case, our algorithm finds the min substring of w to match p.