I collaborated with:

Problem
Please answer Question 17 from Chapter 4 of your text. Try to shoot for an $O(n^2)$ algorithm; it's a bit of a challenge to beat that (although if you're feeling really ambitious, $O(n \log n)$ time is possible...).
Solution

Algorithm 1 Determines whether w contains the pattern p

```
for every interval $i$ do
    run Single Source Scheduling algorithm from class starting on $i$
    keep track of the largest solution out of the Single Source Scheduling runs
end for
```

This algorithm runs in $O(n^2)$ as there are n intervals and the Single Source Scheduling algorithm takes $O(n)$. $O(n) * O(n) = O(n^2)$
Space: Single Source Scheduling creates n solution that take $O(n)$ space so it takes $O(n^2)$.
Suppose there is an optimal solution S consisting of the maximum set of intervals. S starts with interval i, and the Single Source Scheduling algorithm will eventually start with i and determine it is the largest solution. Thus, the output of our solution will be at least as large as S.