- For the approach of always selecting the compatible activity that overlaps the fewest other remaining activities:

i	1	2	3	4	5	6	7	8	9	10	11
s_i	0	1	1	1	2	3	4	5	5	5	6
f_i	2	3	3	3	4	5	6	7	7	7	8
# of overlapping activities	3	4	4	4	4	2	4	4	4	4	3

  This approach first selects a_6, and after that choice it can select only two other activities (one of a_1, a_2, a_3, a_4 and one of a_8, a_9, a_{10}, a_{11}). An optimal solution is $\{a_1, a_5, a_7, a_{11}\}$.

- For the approach of always selecting the compatible remaining activity with the earliest start time, just add one more activity with the interval $[0, 14)$ to the example in Section 16.1. It will be the first activity selected, and no other activities are compatible with it.

Solution to Exercise 16.2-2

The solution is based on the optimal-substructure observation in the text: Let i be the highest-numbered item in an optimal solution S for W pounds and items $1, \ldots, n$. Then $S' = S - \{i\}$ must be an optimal solution for $W - w_i$ pounds and items $1, \ldots, i-1$, and the value of the solution S is v_i plus the value of the subproblem solution S'.

We can express this relationship in the following formula: Define $c[i, w]$ to be the value of the solution for items $1, \ldots, i$ and maximum weight w. Then

$$c[i, w] = \begin{cases} 0 & \text{if } i = 0 \text{ or } w = 0 , \\ c[i-1, w] & \text{if } w_i > w , \\ \max(v_i + c[i-1, w-w_i], c[i-1, w]) & \text{if } i > 0 \text{ and } w \ge w_i . \end{cases}$$

The last case says that the value of a solution for i items either includes item i, in which case it is v_i plus a subproblem solution for $i-1$ items and the weight excluding w_i, or doesn't include item i, in which case it is a subproblem solution for $i-1$ items and the same weight. That is, if the thief picks item i, he takes v_i value, and he can choose from items $1, \ldots, i-1$ up to the weight limit $w - w_i$, and get $c[i-1, w-w_i]$ additional value. On the other hand, if he decides not to take item i, he can choose from items $1, \ldots, i-1$ up to the weight limit w, and get $c[i-1, w]$ value. The better of these two choices should be made.

The algorithm takes as inputs the maximum weight W, the number of items n, and the two sequences $v = \langle v_1, v_2, \ldots, v_n \rangle$ and $w = \langle w_1, w_2, \ldots, w_n \rangle$. It stores the $c[i, j]$ values in a table $c[0..n, 0..W]$ whose entries are computed in row-major order. (That is, the first row of c is filled in from left to right, then the second row, and so on.) At the end of the computation, $c[n, W]$ contains the maximum value the thief can take.

```
DYNAMIC-0-1-KNAPSACK(v, w, n, W)
for w ← 0 to W
    do c[0, w] ← 0
for i ← 1 to n
    do c[i, 0] ← 0
       for w ← 1 to W
           do if w_i ≤ w
                 then if v_i + c[i − 1, w − w_i] > c[i − 1, w]
                          then c[i, w] ← v_i + c[i − 1, w − w_i]
                          else c[i, w] ← c[i − 1, w]
                 else c[i, w] ← c[i − 1, w]
```

The set of items to take can be deduced from the c table by starting at $c[n, W]$ and tracing where the optimal values came from. If $c[i, w] = c[i-1, w]$, then item i is not part of the solution, and we continue tracing with $c[i-1, w]$. Otherwise item i is part of the solution, and we continue tracing with $c[i-1, w-w_i]$.

The above algorithm takes $\Theta(nW)$ time total:

- $\Theta(nW)$ to fill in the c table: $(n+1) \cdot (W+1)$ entries, each requiring $\Theta(1)$ time to compute.
- $O(n)$ time to trace the solution (since it starts in row n of the table and moves up one row at each step).

Solution to Exercise 16.2-4

The optimal strategy is the obvious greedy one. Starting will a full tank of gas, Professor Midas should go to the farthest gas station he can get to within n miles of Newark. Fill up there. Then go to the farthest gas station he can get to within n miles of where he filled up, and fill up there, and so on.

Looked at another way, at each gas station, Professor Midas should check whether he can make it to the next gas station without stopping at this one. If he can, skip this one. If he cannot, then fill up. Professor Midas doesn't need to know how much gas he has or how far the next station is to implement this approach, since at each fillup, he can determine which is the next station at which he'll need to stop.

This problem has optimal substructure. Suppose there are m possible gas stations. Consider an optimal solution with s stations and whose first stop is at the kth gas station. Then the rest of the optimal solution must be an optimal solution to the subproblem of the remaining $m - k$ stations. Otherwise, if there were a better solution to the subproblem, i.e., one with fewer than $s - 1$ stops, we could use it to come up with a solution with fewer than s stops for the full problem, contradicting our supposition of optimality.

This problem also has the greedy-choice property. Suppose there are k gas stations beyond the start that are within n miles of the start. The greedy solution chooses the kth station as its first stop. No station beyond the kth works as a first stop, since Professor Midas runs out of gas first. If a solution chooses a station $j < k$ as