(a) This algorithm is too short-sighted; it might take a high-stress job too early and an even better one later.

	Week 1	Week 2	Week 3
ℓ	2	2	2
h	1	5	10

The algorithm in (a) would take a high-stress job in week 2, when the unique optimal solution would take a low-stress job in week 1, nothing in week 2, and then a high-stress job in week 3.

(b) Let $OPT(i)$ denote the maximum value revenue achievable in the input instance restricted to weeks 1 through i. The optimal solution for the input instance restricted to weeks 1 through i will select *some* job in week i, since it's not worth skipping all jobs — there are no future high-stress jobs to prepare for. If it selects a low-stress job, it can behave optimally up to week $i-1$, followed by this job, while if it selects a high-stress job, it can behave optimally up to week $i-2$, followed by this job. Thus we have justified the following recurrence.

$$OPT(i) = \max(\ell_i + OPT(i-1), h_i + OPT(i-2)).$$

We can compute all OPT values by invoking this recurrence for $i = 1, 2, \ldots, n$, with the initialization $OPT(1) = \max(\ell_1, h_1)$. This takes constant time for each value of i, for a total time of $O(n)$. As usual, the actual sequence of jobs can be reconstructed by tracing back through the set of OPT values.

An alternate, but essentially equivalent, solution is as follows. We define the following sub-problems. Let $L(i)$ be the maximum revenue achievable in weeks 1 through i, given that you select a low-stress job in week i, and let $H(i)$ be the maximum revenue achievable in weeks 1 through i, given that you select a high-stress job in week i.

Again, the optimal solution for the input instance restricted to weeks 1 through i will select some job in week i. Now, if it selects a low-stress job in week i, it can select anything it wants in week $i-1$; and if it selects a high-stress job in week i, it has to sit out week $i-1$ but can select anything it wants in week $i-2$. Thus we have

$$L(i) = \ell_i + \max(L(i-1), H(i-1)),$$

$$H(i) = h_i + \max(L(i-2), H(i-2)).$$

The L and H values can be built up by invoking these recurrences for $i = 1, 2, \ldots, n$, with the initializations $L(1) = \ell_1$ and $H_1 = h_1$.

[1]ex695.414.330