Question 1

Wegmans, the leader in cutting-edge food technology, has hired you to revamp their produce acquisition scheduler. This may sound scary, but it turns out all they want to know is where they should buy their apples each week. And really, they only have 2 choices; they can get apples from the local farmers, or from a large food wholesaler.

The local farmers’ apple supply is dependent on the season, so their prices change on a week by week basis. (Luckily, after many years in the business you have good estimates for these prices). Alternatively, there are the large food wholesalers, who can draw on their many and distant apple producing regions to guarantee constant prices year round. A slight drawback is that these guys only deal with big buyers, and hence are only willing to supply Wegmans in blocks of 4 consecutive weeks. So you can order apples from the wholesaler any week, but you’ll be buying your next 4 weeks of apples from them. If you want more than 4 consecutive weeks, it had better be 8, or 12, etc.

Formally, we would like to have a buying schedule for the next \( n \) weeks. The cost of buying apples from local farmers on week \( i \) is \( c_i \). The cost of buying apples from the wholesalers for any 4 week period of time is \( c' \). For simplicity, assume that you can not place an order from the wholesalers for the 4 week period starting with week \( n-2 \), \( n-1 \), or \( n \). We would like to find the cheapest way of buying apples for all \( n \) weeks.

For example, if the local prices of apples are 4,8,7,5,3,7,9,4,7,9, and \( c' = 16 \) then you should buy locally during weeks 1, 6, and 7, and buy from the wholesalers for weeks 2 through 5 and weeks 8 through 11. The total cost of this plan is 46.

a) Consider the following greedy algorithm: Consider the weeks in order, beginning with week 1. When looking at week \( i \), compare \( c_i + c_{i+1} + c_{i+2} + c_{i+3} \) with \( c' \). If \( c' \) is larger, buy locally on week \( i \) and move on to week \( i+1 \). Otherwise, buy from the wholesaler for the weeks \( i \) through \( i+3 \) and move on to week \( i+4 \).

Give an example where this algorithm fails to find the optimal solution.
b) Give an algorithm to solve this problem, and prove correctness. The algorithm should run in time polynomial in \( n \).

Question 2

You have finally received the patent for your fingernail-sized supercomputer, along with the construction process needed to churn ’em out. Sadly, you have neither the skills nor the tools to actually build these little suckers, and all the local nanotech labs, being as specialized as they are, would probably charge an arm and a leg to make them.

Fortunately, the process for making your tiny computers is fairly modular, and can be thought of as a number of distinct sequential steps. It turns out that each individual step can be completed cheaply by at least one of the many nanotech labs. You figure that maybe by breaking your design process up into smaller pieces, these pieces can then be farmed out to various labs (different labs charge different prices for different tasks). Of course, you may not want to break up your process completely, as you will undoubtedly incur some overhead in moving your product around.

Lets make things a little more precise. You have \( n \) ordered steps in your process. You also have a function \( f(\cdot, \cdot) \), where \( f(i, j) \) is the lowest price any single lab charges to carry out steps \( i \) through \( j \) (\( f(i, j) \) is only defined for \( i \leq j \)). Furthermore, there is a constant cost \( c \) incurred every time you ship a product to a new lab. Give an algorithm that is polynomial in \( n \) and finds the minimum cost of building these machines. And as usual, we also want to see a proof of correctness.

For example, suppose you have 3 steps in your process. Further assume that \( f(1, 1) = 5, f(1, 2) = 7, f(1, 3) = 16, f(2, 2) = 6, f(2, 3) = 8, f(3, 3) = 7 \), and \( c = 2 \).

Then the minimum cost of building one of these computers is 17, since there is a lab which will do the first step for a cost of 5, and another lab which will do steps 2 and 3 together for a cost of 8. Since the product must be delivered to 2 labs, you will pay an additional 4 in shipping, for a total of 17. There are 3 other possible solutions, each of which is more expensive. Remember that your algorithm need only return the final cost (in this case 17).
Question 3

You have recently been hired as a consultant for Texon, a large fuel and oil company which is seeking to expand its chain of gas stations into upstate New York. It turns out that the recent success of the H2 Hummer has roughly doubled their yearly revenues. As a result, they have enough cash lying around to open up $k$ gas stations on Interstate 90 between Albany and Buffalo.

As you might have suspected, it turns out that there are $n$ rest areas where gas stations can be built on I90. Your employer has already come up with estimates $p_i$ of how much profit would be generated by building a gas station at rest area $i$, for all $i$. Unfortunately, due to recent anti-monopolistic regulations, no two adjacent rest areas can both have a Texon gas station.

Your employer would like an efficient algorithm that picks a set $S$ of rest areas where Texon should build gas stations so as to maximize

$$\sum_{i \in S} p_i$$

subject to the following two constraints:

$|S| \leq k$, and

If $i \in S$ then $i + 1 \notin S$.

a) One of your partners has the following plausible idea: Run the algorithm given in class for finding the max weight independent set of points on a path, and then drop all but the $k$ most profitable sites. Give an example where this algorithm fails to find the optimal solution.

b) Give an algorithm to solve the problem. Your algorithm should run in time polynomial in $n$ and $k$. Prove that your algorithm is correct and give a brief analysis of its running time.

**Hint:** Think about dynamic programming with an additional variable.