Homework 3 (CS 401)
Deadline: **Wednesday April 6, 11:59 pm CST**

1. 20 pts

Suppose you are managing a consulting team of expert computer hackers, and each week you have to choose a job for them to undertake. The set of possible jobs is divided into those that are low-stress (e.g. setting up a Web site for a class at the local elementary school) and those that are high-stress (e.g., protecting the dean's most valuable secrets.) The basic question each week is whether to take on a low-stress job or a high-stress job. If you select a low-stress job for your team in week i, then you get a revenue of $l_i > 0$ dollars; if you select a high-stress job, you get a revenue of $h_i > 0$ dollars. High-stress jobs typically pay more. The catch, however, is that in order for the team to take on a high-stress job in week i, it is required that they do no job (of either type) in week $i - 1$; they need a full week of prep time to get ready for the crushing stress level. On the other hand, it is okay for them to take a low-stress job in week i even if they have done a job (of either type) in week $i - 1$. So, given a sequence of n weeks, a plan is specified by a choice of low-stress, high-stress or none for each of the n weeks, with the property that if high-stress is chosen for week $i > 1$, then none has to be chosen for week $i - 1$. (It is okay to choose a high-stress job in week 1.) The value of the plan is determined in the natural way; for each i, you add li to the value if you choose low-stress in week i, and you add hi to the value if you choose high-stress in week i. (You add 0 if you choose none in week i.) Given sets of values $l_1, l_2, \dots, l_n$ and $h_1, h_2, \dots, h_m$, find a plan of maximum value (such a plan is called optimal.)

Example. Suppose $n = 4$, and the values of i and h_i are given by the following table. Then the plan of maximum value would be to choose "none" in week 1, a high-stress job in week 2, and low-stress jobs in weeks 3 and 4. The value of this plan would be $0 + 50 + 10 + 10 = 70$.

	Week 1	Week 2	Week 3	Week 4
l	10	1	10	10
h	5	50	5	10

(a) Find the dynamic programing formulation for the problem. (10 pts)

(b) Provide the efficient algorithm based on your formulation. (10 pts)

2. 30 pts

The residents of the underground city of Zion defend themselves through a combination of kung fu, heavy artillery, and efficient algorithms. Recently they have become interested in automated methods that can help fend off attacks by swarms of robots. Here's what one of these robot attacks looks like.

• A swarm of robots arrives over the course of n seconds; in the ith second, x_i robots arrive. Based on remote sensing data, you know this sequence $x_1, x_2, \dots, x_n$ in advance.

• You have at your disposal an electromagnetic pulse (EMP), which can destroy some of the robots as they arrive; the EMP's power depends on how long it's been allowed to charge up.

■ Abolfazl Asudeh

To make this precise, there is a function $f(\bullet)$ so that if j seconds have passed since the EMP was last used, then it is capable of destroying up to $f(j)$ robots.

• So specifically, if it is used in the kth second, and it has been j seconds since it was previously used, then it will destroy $min(x_k, f(j))$ robots. (After this use, it will be completely drained.)

• We will also assume that the EMP starts off completely drained, so if it is used for the first time in the jth second, then it is capable of destroying up to $f(j)$ robots.

The problem. Given the data on robot arrivals $x_1, x_2, \ldots, x_n$, and given the recharging function $f(\bullet)$, choose the points in time at which you're going to activate the EMP so as to destroy as many robots as possible.

Example. Suppose $n = 4$, and the values of x_t and $f(i)$ are given by the following table.

i	1	2	3	4
x_i	1	10	10	1
$f(i)$	1	2	4	8

In the above example, the optimal solution is to activate EMP in 3rd and 4th seconds. In 3rd second the EMP has been charged for 3 seconds and destroys $\min(4,10) = 4$ robots. In 4th second, it has charged for 1 seconds and destroys $\min(f(1), x_4) = 1$ robot. This is a total of 5.

(a) Show that the following algorithm does not correctly solve the problem by proving a contradictory example. (10 pts)

```
Schedule-EMP(x_1, ..., x_n)
  Let j be the smallest number for which f(j) ≥ x_n
    (If no such j exists then set j = n)
  Activate the EMP in the n^th second
  If n − j ≥ 1 then
    Continue recursively on the input x_1, ..., x_{n−j}
      (i.e., invoke Schedule-EMP(x_1, ..., x_{n−j}))
```

In your example, say what the correct answer is and also what the algorithm above finds.

(c) Give the dynamic programing formulation and the algorithm that takes the data on robot arrivals$x_1, x_2, \ldots, x_n$, and the recharging function $f(\bullet)$, and returns the maximum number of robots that can be destroyed by a sequence of EMP activations. (10 pts formulation, 10 pts algorithm)

Homework 3 (CS 401)
Deadline: **Wednesday April 6, 11:59 pm CST**

3. 20 pts

The problem of searching for cycles in graphs arises naturally in financial trading applications. Consider a firm that trades shares in n different companies. For each pair $i \neq j$, they maintain a trade ratio r_{ij}, meaning that one share of i trades for r_{ij} shares of j. Here we allow the rate r to be fractional; that is, $r_{ij} = 2/3$ means that you can trade three shares of i to get two shares of j. A trading cycle for a sequence of shares $i_1, i_2 i_3, \ldots, i_k$ consists of successively trading shares in company i_1 for shares in company i_2, then shares in company i_2 for shares i_3, and so on, finally trading shares in i_k back to share sin company i_1. After such a sequence of trades, one ends up with shares in the same company i_1 that one starts with. Trading around a cycle is usually a bad idea, as you tend to end up with fewer shares than you started with. But occasionally, for short periods of time, there are opportunities to increase shares. We will call such a cycle an opportunity cycle, if trading along the cycle increases the number of shares. This happens exactly if the product of the ratios along the cycle is above 1. In analyzing the state of the market, a firm engaged in trading would like to know if there are any opportunity cycles.

Give a polynomial-time algorithm that finds such an opportunity cycle, if one exists.

4. 30 pts

Given two matrices $A_{n\times m}$ and $B_{m\times k}$, consider the algorithm that computes $A \times B$ using (nmk) scalar multiplications. Suppose you are given sequence of matrices $A^{(0)}_{x_0\times x_1}$, $A^{(2)}_{x_1\times x_2}, \ldots, A^{n}_{x_{n-1}\times x_n}$ to multiply. That is, you want to compute $A^{(1)} \times A^{(2)} \times \ldots \times A^{(n)}$. Note that different ordering of conducting the operations have different costs. For example, consider a set of 4 matrices with the dimensions specified with the following array:

i	0	1	2	3	4
x_i	10	30	5	15	5

In this example, if the operations are done as $(((A^{(1)} \times A^{(2)}) \times A^{(3)}) \times A^{(4)})$ –/* multiply $A^{(1)}$ and $A^{(2)}$, then multiply the result with $A^{(3)}$, and multiply its result with $A^{(4)}$ */-- the cost is equal to $10 \times 30 \times 5 + 30 \times 5 \times 15 + 5 \times 15 \times 5 = 4125$, while if the order of operations was $((A^{(1)} \times A^{(2)}) \times (A^{(3)} \times A^{(4)}))$, it would be $10 \times 30 \times 5 + 5 \times 15 \times 5 + 10 \times 5 \times 5 = 2125$.

The problem. Given an array x (of size $n + 1$) specifying the dimensions of a sequence of n matrices to multiply, we want to find the ordering of conducting operations with minimum cost.

(a) Write the dynamic programing formulation of the solution. (15 pts)
(b) Provide the efficient algorithm for solving the problem. (15 pts)

■ Abolfazl Asudeh