cs4800 - algorithms - s'18 — nguyen
due fri feb 9, 2018 at 1:35pm via gradescope

H4.

- The assignment is due at Gradescope on February 9 at 1:35pm. Late assignments will not be accepted. Submit early and often.
- You are permitted to study with friends and discuss the problems; however, *you must write up you own solutions, in your own words*. Do not submit anything you cannot explain. If you do collaborate with any of the other students on any problem, please do list all your collaborators in your submission for each problem.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly prohibited.
- We require that all homework submissions are prepared in Latex. If you need to draw any diagrams, however, you may draw them with your hand.

problem 1 *Planet Laser*

The NASA Near Earth Object Program lists potential future Earth impact events that the JPL Sentry System has detected based on currently available observations. Sentry is a highly automated collision monitoring system that continually scans the most current asteroid catalog for possibilities of future impact with Earth over the next 100 years.

This system allows us to predict that i years from now, there will be x_i tons of asteroid material that has near-Earth trajectories. In the mean time, we can build a space laser that can blast asteroids. However, each laser blast will require *exajoules* of energy, and so there will need to be a recharge period on the order of *years* between each use of the laser. The longer the recharge period, the stronger the laser blast; e.g. after j years of charging, the laser will have enough power to obliterate d_j tons of asteroid material. This problem explores the best way to use such a laser.

The input to the algorithm consists of the vectors $(x_1, \ldots, x_n)$ and $(d_1, \ldots, d_n)$ representing the incoming asteroid material in years 1 to n, and the power of the laser d_i if it charges for i years. The output consists of the optimal schedule for firing the laser which obliterates the most material.

Example Suppose $(x_1, x_2, x_3, x_4) = (1, 10, 10, 1)$ and $(d_1, d_2, d_3, d_4) = (1, 2, 4, 8)$. The best solution is to fire the laser at times 3, 4 in order to blast 5 tons of asteroids (4 tons at time 3 and 1 ton at time 4).

(a) Construct an instance of the problem on which the following "greedy" algorithm returns the wrong answer:

> **procedure** BadLaser($(x_1, \ldots, x_n), (d_1, \ldots, d_n)$)
>   Compute the smallest j such that $d_j \geq x_n$. Set $j = n$ if no such j exists.
>   Shoot the laser at time n.
>   **if** $n > j$ **then return** BadLaser($(x_1, \ldots, x_{n-j}), (d_1, \ldots, d_{n-j})$)
>   **end if**
> **end procedure**

Intuitively, the algorithm figures out how many years (j) are needed to blast all the material in the last time slot. It shoots during that last time slot, and then accounts for the j years required to recharge for that last slot, and recursively considers the best solution for the smaller problem of size $n - j$.

Solution:

(b) Let $\textsc{best}_j$ be the maximum amount of asteroid we can blast from year 1 to year j. Give a recurrence to compute $\textsc{best}_j$ from $\textsc{best}_1, \ldots, \textsc{best}_{j-1}$. Justify your recurrence.

Solution:

(c) Describe a dynamic programming algorithm based on your recurrence above, including the base cases and order of evaluation. Analyze the running time of your solution.

Solution:

PROBLEM 2 *Bitcoin trader*

As a cash-strapped Northeastern student, Alice decided to get rich by trading bitcoin. Bitcoins are traded everyday. Prices vary wildly across different days but on any single day the price is exactly the same. Alice's strategy is to buy bitcoins when the price is low and sell them when the price is high. Putting all her lunch money together, Alice can afford roughly 0.1 bitcoin and she decided that at any point in time, she will have either 0.1 bitcoin or 0 bitcoin. Everyday, Alice can either do nothing or change her holdings from 0 bitcoin to 0.1 or vice versa. Furthermore, every bitcoin transaction will cost Alice \$1. As a CS-finance double major, Alice makes perfect predictions of bitcoin prices for the next n days, which she stores in an array $p[1, \ldots, n]$. Please help her find the perfect sequence of trades to make the most amount of money.

For instance, suppose the bitcoin prices for $n = 8$ days are $13, 7, 8, 9, 13, 11, 10, 16$. Alice can buy on day 2 and sell on day 5, making $13 - 7 - 2$(transaction costs) $= 4$. Then she can buy on day 7 and sell on day 8, making $16 - 10 - 2$(transaction costs) $= 4$. Her total earnings are 8.

(a) Let $Have(i)$ be Alice's maximum total earnings from days 1 to i provided that she has 0.1 bitcoin at the end of day i. Let $NotHave(i)$ be Alice's maximum total earnings from days 1 to i provided that she has 0 bitcoin at the end of day i. Give a recurrence for computing $Have(i)$ and $NotHave(i)$. Hint: $Have(i)$ depends on both $Have(i-1)$ and $NotHave(i-1)$.

Solution:

(b) Give a dynamic programming algorithm to compute Alice's maximum total earnings using your recurrence. Analyze the running time of your algorithm.

Solution: