CS 577: Introduction to Algorithms	Homework 2
Out: 02/09/21	**Due: 02/16/21**

Name: ______________________ Wisc ID: ______________________

Ground Rules

- Answer the questions in the boxes provided on the question sheets. If you run out of room for an answer, add a page to the end of the document.
- The homework is to be done and submitted individually. You may discuss the homework with others in either section but you must write up the solution *on your own*.
- You are not allowed to consult any material outside of assigned textbooks and material the instructors post on the course websites. In particular, consulting the internet will be considered plagiarism and penalized appropriately.
- The homework is due at 11:59 PM CST on the due date. No extensions to the due date will be given under any circumstances.
- Homework must be submitted electronically on Gradescope.

Problem

You have a backpack that can hold up to k pounds and can be filled with pixie dust or dragon scales. The value of i pounds of pixie dust is determined by the non-decreasing sequence a_i, while the value of j pounds of dragon scales is determined by the non-decreasing sequence b_j. Given a weight limit k and sequences $a_0 \ldots a_n$ and $b_0 \ldots b_n$, your goal is to calculate the maximum value you can obtain by filling the backpack with some combination of pixie dust and dragon scales.

Note that in some cases the rate of increase in an item's value decreases the more you have of it. Such a sequence s_i is called *concave* and satisfies that $s_i - s_{i-1}$ is a decreasing function of i.

Part 1:

For the first part of this question we will assume that both sequences a and b are concave.

(a) Prove that for any k, the sequence $y_i = a_i + b_{k-i}$ for $0 \leq i \leq k$ is concave.

(b) The maximum total value of the items in your backpack defines a sequence v_k as a function of the capacity k, where $v_k = \max_{i \in [0,k]} a_i + b_{k-i}$. Provide a divide-and-conquer algorithm based solution to compute v_k which has a running time of $O(\log k)$ along with a brief (2-3 lines) proof of correctness.

Part 2:

In the second part we will consider the scenario where only sequence *b* is concave and *a* is not.

Note: In such a situation computing v_k for a fixed value of k takes $\Omega(k)$ time in the worst-case. This can be seen by noting that the corresponding function $y_i = a_i + b_{k-i}$ can result in an arbitrary unsorted sequence and that finding the maximum element in an unsorted list of length k takes $\Omega(k)$ time as one needs to look at all the elements.

However, even though it takes $\Omega(k)$ time to compute the value of v_k, you will show that the whole sequence for all $k \in [0, n]$ can be computed significantly faster than the naive algorithm which runs in $O(n^2)$ time.

(a) Defining $i(k)$ as the highest index that maximizes the value $a_i + b_{k-i}$, prove that $i(k)$ is a non-decreasing function of k, i.e. $i(k) \leq i(k+1)$, when b is a concave sequence but a is not.

(b) Provide a divide and conquer algorithm to compute v_k for $k = 0, 1, \ldots n$ that runs in time $O(n \log n)$ along with a brief (2-3 lines) proof of correctness. You may assume that n is a power of 2 for simplicity.

Part 3:

Provide an algorithm to compute v_k for all $k = 0, 1, \dots, n$ running in time $O(n^{1.99})$ or faster.

This is a major open problem in computer science. There is no existing solution and we do not expect you to come up with one.