CS 577: Introduction to Algorithms | Homework 2

Out: 02/09/21 | Due: 02/16/21

Name: Aidan Tepper Wisc ID: 9081242969

## Ground Rules

- Answer the questions in the boxes provided on the question sheets. If you run out of room for an answer, add apage to the end of the document.
- The homework is to be done and submitted individually. You may discuss the homework with others in eithersection 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 thecourse 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 anycircumstances.
- 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 $a_0 \dots a_n$ and $b_0 \dots b_n$, 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.

WTS $\forall i$, $y_i - y_{i-1} \geq y_{i+1} - y_i$

$$y_i - y_{i-1} = (a_i + b_{k-i}) - (a_{i-1} + b_{k-i+1})$$
$$\geq (a_{i+1} - a_i) + (b_{k-i-1} - b_{k-i})$$
$$= (a_{i+1} + b_{k-i-1}) - (a_i + b_{k-i})$$
$$= y_{i+1} - y_i$$

$\therefore y_i$ 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.

Input: two concave and non-decreasing series $a_0 \ldots a_n$ and $b_0 \ldots b_n$, and an integer $k$ such that $0 \le k \le n$

Output: $v_k$, the max value of $y_i = a_i + b_{k-i}$ for $0 \le i \le k$.

if less than 3 elements, find max by comparing values.

using index $j = \lfloor k/2 \rfloor$, look at center value, $j+1$, and $j-1$. If max value is $y_j$, return $y_j$.

If max is $y_{j-1}$, recursively call with only elements left of $y_j$.

If max is $y_{j+1}$ " only elements left ".

Because $y_i$ is concave, looking at the middle element and the left/right elems provides sufficient info to determine whether the max is either in the center, to the left, or to the right. The algorithm will then recurse on the correct side and eventually find the max (by induction). Each time it recurses, it narrows down until it terminates at the proper maximum. ∎

## 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. $i(k) \le i(k+1)$, when $b$ is a concave sequence but $a$ is not.

Given $i(k) \le i(k+1)$, we will show that $i(k) = a_i + b_{k-1}$

$i(k+1) = a_{i+1} + b_{k+1-i}$

$a_i + b_{k-i} \le a_{i+1} + b_{k+1-i}$

$b_{k-i} - b_{k+1-i} \le a_{i+1} - a_i$

Because $b_{k-i} \le b_{k+1-i}$ (it's concave) and $a_{i+1} \ge a_i$ (its convex), then $b_{k-i} - b_{k+1-1}$ at max. can be 0.

Therefore, because $a_{i+1} - a_i \ge 0$, $b_{k-i} - b_{k+1-i} \le a_{i+1} - a_i$ and $i(k) \le i(k+1)$. ∎

(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.

Input: one concave and one non-decreasing sequence

Output: $v_k$, the max value of $y_i = a_i + b_{k-i}$ for $k$ such that $0 <= k <= n$

Similar to 1b, look at center ($j = \lfloor k/2 \rfloor$), $j+1$, and $j-1$.

If max out of these 3 elements is $y_i$, return $y_i$.

If $y_{i-1}$, recurse on $y_{i-1}$

If $y_{i+1}$, recurse on $y_{i+1}$

Because $y$ is non-decreasing, considering the center and 2 enclosing elements is sufficient for either determining the max or determining the correct side to search on. The algorithm recurses on the correct side, repeating with smaller number of elements until terminating on the correct maximum element.

## Part 3:

Provide an algorithm to compute $v_k$ for all $k = 0,1,\ldots,n$ 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.*