1.6 Geese Trenchcoat

Andre Monteiro

Term 3, 2025

Problem

There exists a group of geese, of which we must find two that match our height H.

We propose a solution using a "two-pointer" method, essentially sorting then pointing to the first and last entry, slowly iterating both pointers inward, till they either intersect or sum target height H.

Method

(a) **Sort using merge sort, and place red hat R on the shortest, and blue hat B on the tallest**

Time complexity: $O(n \log n)$

Justification: Merge sort splits a large array into smaller pieces before reunifying it at the end. This is an effective way of reordering the array in ascending order, and is neccessary to correctly be able to match heights. As merge sort doesn't insert or delete items, all solutions are still present.

(b) **If the hatted geese have total height less than H, move the red hat one space to the right**

Time complexity: $O(n)$

Justification: Time complexity is $O(n)$, as at worst, the hat will move all the way to the right before returning no solution. Since iteratively moving right does not delete or skip over items, this implies that no solution will be missed. Furthermore, this case requires $R + B < H$, and as all geese to the left of R are shorter, there exists no solution that would generate a larger number as required.

(c) **If the hatted geese have total height greater than H, move the blue hat one space to the left**

Time complexity: $O(n)$

Justification: Time complexity is $O(n)$, as at worst, the hat will move all the way to the left before returning no solution. Again, since iteratively moving left does not delete or skip over items, this implies that no solution will be missed. Furthermore, this case requires $R + B > H$, and as all geese to the right of B are taller, there exists no solution that would generate a smaller number as required.

(d) **If the hatted geese have total height equal to H, report this pair as a solution. If a single goose ever has both hats, report that there is no solution**

Time complexity: $O(1)$

Justification: Time complexity is $O(1)$, as this is a simple comparison check. If the hats ever intersect, this implies that there exists no two geese heights within the group that add to the target number H. This is as upon intersection, the previous geese combination was either

$R + B > H$ OR $R + B < H$. The hat movement will result in a number that is inverse of this condition, e.g. if it was $R + B < H$, incrementing the geese number will result in $R + B > H$, effectively after crossing over, B will become the new R and vis versa, which were cases we previously checked.

Time Complexity Analysis

The algorithm consists of two main phases:

(a) **Sorting:** We first sort the geese heights using merge sort, which takes $O(n \log n)$ time. Sorting is necessary to apply the two-pointer method correctly.

(b) **Two-pointer scan (hat movements):** We place a red hat on the shortest goose and a blue hat on the tallest. At each step, we move either the red hat one step right (if the sum is too small) or the blue hat one step left (if the sum is too large). Each movement reduces the number of candidate pairs by one.
Justification: Although each hat can move multiple times, the total number of movements of both hats combined is at most n, so this phase runs in $O(n)$ time.

Overall complexity: Combining the two phases and termination, the total time is

$$O(n \log n) + O(n) + O(1) = O(n \log n),$$

As the term $O(n \log n)$ asymptomatically dominates the rest, the final complexity is

$$\boxed{O(n \log n)}.$$