COMP3121-Ass2-Q3

z5302513, Kihwan Baek

July 2021

We first make the array of pairs (a_i, g_i) for all a_i and g_i and set the index from i to n. After that, we use merge sort for the array in ascending based on a_i value and if the a_i and a_{i+1} are the same, compare the g_i and g_{i+1} and put the larger one before the smaller one. It takes $O(N \log N)$ time.

From the first pair of (a_i, g_i) which is (a_1,g_1), we update the initial strength to S-a_1 and check if the updated strength is less than 0. If it is less than 'zero', we can say there is "no such ordering". Otherwise, we update the strength to $updatedS + g_1$. Using this process, we check for all pairs (a_i, g_i) and it takes $O(N)$time. Finally, if this algorithm finishes without "no such ordering", we have an algorithm which determines in which order the hero can fight the monsters if he is to kill them all and the total time complexity is $O(N \log N + N) = O(N \log N)$.