

# Analysis: Shifts

## Test set 1 (Visible)

For each shift, we have three choices,

1. Aninda alone guards the shift
2. Boon-Nam alone guards the shift
3. Both of them guard the shift

We can go through all possible choices for all shifts, and check for how many combinations both of their happiness score is at least  $H$ . There are total  $3^N$  such combinations possible, enumerating through all of them fits the time limit for test set 1.

## Test set 2 (Hidden)

For test set 2, we can divide the shifts into 2 sets, each having the size of at most  $\text{ceil}(N/2)$ . We can divide them into two sets, because the choices for each shift are independent of each other. For each set, we can enumerate every possible combination, and compute happiness score for both of the guards for each combination, which gives us a list of happiness score pairs for those two sets. Then for each happiness score pair  $(h_1, h_2)$  of the combinations of set 1, we need to count the number of happiness score pairs  $(p_1, p_2)$  of the combinations of set 2, such that  $p_1 + h_1 \geq H$  and  $p_2 + h_2 \geq H$ , which can be converted to  $p_1 \geq H - h_1$  and  $p_2 \geq H - h_2$ .

Essentially, we can follow these steps below to find the answer.

1. Store the happiness score pairs of combinations of both set 1 and set 2 in two arrays, let's call them  $A_1$  and  $A_2$ .
2. Sort both  $A_1$  and  $A_2$ , by the first value in the pair in decreasing order, then by the second value of the pair in decreasing order.
3. Iterate through  $A_1$ . For each pair  $(h_1, h_2)$  in  $A_1$ , do the following:
  - Find the pairs  $(p_1, p_2)$  in  $A_2$  where  $p_1$  is not less  $H - h_1$ .
  - For each of the pair  $(p_1, p_2)$  found, add  $p_2$  to a data structure.
  - Find the number of elements not less than  $H - h_2$  in the data structure, add that to answer.

For this approach, we need a data structure that supports these two operations:

1. Insert a number in the data structure
2. Count number of elements in the data structure not less than a given integer

These two operations can be done in  $O(\log_2(X))$  time using [range tree](#), where  $X$  is the number of elements in the data structure. Here upper bound of  $X$  is  $3^{(N/2)}$ . The total complexity of this approach will be  $O(3^{(N/2)} \times \log_2(3^{(N/2)})) \sim O(3^{(N/2)} \times N)$  which is sufficient for test set 2.