

# Analysis: Toys

Let us say Axel can play with the toys in multiple rounds (starting from 1), where each round starts with the toy that has the smallest number and finishes with the toy that has the largest number. There are a few observations:

1. Axel can play with each toy at least once, in round 1.
2. If Axel cannot play indefinitely, in round 2 he will get stuck at some toy and start crying.
3. If Axel can play with a toy more than twice (i.e. reaches round 3), he can keep playing indefinitely.

## Test Set 1

For this test set, we can generate all subsets of the toys in order. For each subset, we can loop over the toys in the subset and calculate the maximum time Axel can play. This maximum time will be either INDEFINITELY (if Axel can play more than twice as in Observation 3), or sum of the enjoyments of the toys played until Axel gets stuck as in Observation 2. We can keep track of the maximum time for each subset and if two subsets have the same maximum time, we will consider the one with minimum toys removed (maximum subset size). As number of subsets will be  $2^N$  and for each subset, time spent by Axel is calculated in linear time, total complexity of the solution will be  $O(N \times 2^N)$ .

## Test Set 2

Let us say we have  $K$  toys. If Axel can play indefinitely with these toys, for each toy, its remembrance should be less than or equal to the sum of enjoyment of all other toys except this one. If  $SUM$  is total sum of enjoyments of all the toys that Axel can play with, we can say, for each  $i = 1$  to  $K$ ,  $R_i \leq SUM - E_i$  Or  $R_i + E_i \leq SUM$ .

If Axel cannot play with these  $K$  toys indefinitely, we can try removing all the toys violating the condition  $R_i + E_i \leq SUM$ , so that we can get a list of toys with no violation and Axel can play indefinitely. Here, we can remove the toys violating the condition in any order. But for simplicity, we will first remove a toy for which  $R_i + E_i$  is the largest. The reason being, if we remove some other toy, it will only decrease the  $SUM$  by enjoyment of the other toy and this toy would still be violating the condition  $R_i + E_i \leq SUM$ .

In this test set, we can keep a list of the toys such that Axel can play with them indefinitely. Also, we will have a track of total time played till now (let us say *cur\_time*), maximum possible time that Axel can spend playing (let us say *max\_time*) and count of toys removed. Initially, *cur\_time* will be total time taken in round 1 i.e.  $SUM$  as per Observation 1. Now, we will simulate the round 2 and have an empty list initially. We will keep adding toys 1 to  $N$  one by one to this list. After adding a toy to the list, we will check if we have some toy in the list violating the condition  $R_i + E_i \leq SUM$ . If so, we will remove the toy violating the condition that has the largest  $R_i + E_i$ . Once we add a toy to the list, we will add its enjoyment to *cur\_time* and while removing a toy from the list, we will remove its corresponding time (i.e. twice the enjoyment time for this toy) from *cur\_time*, and remove its enjoyment time from the  $SUM$ . After each toy is added and toys violating the condition are removed, we will update *max\_time* if *cur\_time* is optimal. Also, we will keep track of total toys removed from the list and update that as well while updating *max\_time*. Finally, when all the toys have been added to the list and processed, if the list is not empty, Axel can play with the toys indefinitely. Else, we have *max\_time* and corresponding toys removed as our answer.

To maintain the list of toys and remove a toy with the largest  $R_i + E_i$  efficiently, we can use [Priority Queue](#) to store toys in decreasing order of their  $R_i + E_i$ . As we are processing each toy twice, once for calculating initial *SUM*, and other for processing second round where in priority queue each toy can be added/removed maximum one time, total complexity of solution will be  $O(N \log N)$ .