**Exchange arguments**

If we are given items and we should choose some of them and choose their order, we should sort them with some strange or tricky comparator and then do dynamic programming.

: best possible result if we chose items in the prefix .

The comparator can be found by solving .

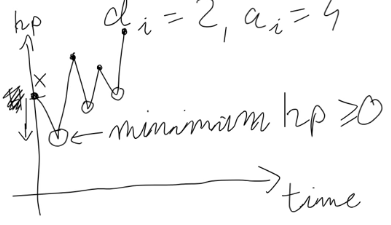
bad comparator:

Example 1: Boxes

There are boxes, each with weight and strength . Choose a subset of boxes and rearrange them in a particular order. Find the maximum possible number of boxes in a tower where the strength of a box must not be smaller than the sum of weights above.

Example 2: Hero

You have health and monsters. Each monster has damage and drops restoration potion after death. Find if you can choose the order of fights such that your health never drops below . If yes, also find the order of the monsters.

minimize the distance between starting point and the lowest point, check if distance is smaller than .

Example 3: Farmcraft

You are given a rooted tree with vertices. You are at the root and moving along an edge takes 1 minute. You will visit all vertices and get back to the root as fast as possible, or minutes. When we visit vertex for the first time, the vertex will be “ready” after minutes. Find the minimum time when all vertices are ready.

Example 4: Pretty Good Proportion

Given a binary string and a double r, find a substring in which the proportion of ones is the closest to r.

Example 5: Different taste

There are cookies, each with and . You start by choosing one and add to your score. Your opponent chooses another one and add to his score. If both players play optimally, who will win?

Maximize

If

Sort by , then check if sum of odd is larger than sum of even

Another interpretation: you choose a cookie, then the opponent blocks one

Example 6: Bear eats

Almost the same as example 5, but the opponent chooses cookie with maximum . are distinct.

Example 7: Coins

There are stacks of coins, each with 2 coins of value and . You can take coins but if you want the bottom coin (), then you must take the top coin () first. What is the maximum value?

If , you can just think of them as separate stacks as if you only need 1 coin, you will take the bigger one, or .

There will at most be 1 stack where you only take the top coin but because if you take 2 stacks, you can replace the smaller with the of the larger as .

Iterate over the 1 special stack, then treat the problem as knapsack. If , make them 2 things with weight 1 and value and . If , make them 1 thing with weight 2 and value .

Sort the things by value divided by weight, and if current size is but the next thing has weight 2, find the minimum of adding the nearest object with weight 1, or not taking the original object with weight 1 and taking the nest object with weight 2.

Example 8: Parenthesis

Given a list of strings with opening and closing parenthesis with total length not exceeding . You can reorder the strings and delete some characters. What is the maximum possible length of the final concatenated string if it needs to be a balanced parenthesis sequence?