

# StockYourBookshelf

Asher Kirshtein

May 5 2022

## 1 My DP Solution and Sub solutions

The way I solved my solution was I first found what my max was and then from there I found what the path to that was.

The way I found the max was I started by iterating through each price in each seforim category. If I had 7 categories that would mean I would have 7 different lists containing each possible amount I could have after adding each price from that category(I decided not to include anything that would take us over our budget because It would be a waste of memory and time to store). Then once I got to my last list I took my max and that is the highest amount I could possibly have.

Once I had my max I was able to take my max, select it from that list. Then I would check my seforim prices from the last category (would be the 6th list in our case) I added and see which numbers I could get to. I then repeated that process until I was just left with the path that could get me to my answer. In some cases I would have multiple paths that could work so I would just choose a random number when I got to a "fork" and then I would do the same thing from there to be left with a correct path. Finally to get my solution list I would take my path and subtract from each step and add the difference into my solution.

## 2 Recursive Formulation

Ex: siddurim = [1, 3, 5] tanach = [2,6,12] machzor = [1,2,4] budget = 8

Finding max:

Step 1: [0]

Step 2: [0] [1, 3, 5]

Step 3: [0] [1, 3, 5] [3, 7, 5]

Step 4: [0] [1, 3, 5] [3, 7, 5] [4, 8, 6, 5, 7]

I would then see that my max is 8 since it is the max in my last list/ list I have after seeing each possibility I can get when adding each value together

Finding solution:

8 is my max so I take that so I have [0] [1, 3, 5] [3, 7, 5] [8]

I then would look back and see which values can get me to 8. The only one is 7 because  $8 - 1 = 7$  I then have [0] [1, 3, 5] [7] [8].

I then see what I can get to 7 from I can either get 1 or 5 since  $7 - 6 = 1$  and  $7 - 2 = 5$  so I can pick either so lets just say 1. I get [0] [1] [7] [8].

I now get the difference of each which is [1, 6, 1].

### 3 My Algorithms Performance

My method which gets the maximum we can get with the budget is  $O(mn^2)$  m is the amount of categories and n is the amount of prices within each category. I need to iterate through each list 1 time to get each row at each step. I then need to get each possible variation from each step which means I go over each n values n times.

My solution method will need my max method to be run first so it will take that amount of time in addition to it's own time constraints it will also take  $O(nm)$  since you only have to go back through the possible routes you already have and take the path you are on at each step If things don't work worst case you will go through  $m * n$  possibilities.