Algorithms Project 3
Tesseract Stacking

1 Introduction

Suppose you live in a universe with four spatial dimensions, you have a set of n 4-Dimensional rectangular blocks where the i^{th} block has dimensions $w(i), x(i), y(i), z(i)$, all of the blocks are of different sizes. You want to place the blocks such that the 3D projection of the current block contains the 3D projection of the block before it so that you maximize the sum of w.[1]

If we constrain the rotations to be such that for the 3D projection of the box, x will be the smallest value, y will be the next smallest, and z is what remains, there are **4** different rotations for the 4D block. Thus we create a list of length $4n$ that holds all the rotations for all the blocks.

1.1 Assumptions

1. You may use the same block more than once
2. The volume of the 3D projection of the block inside must be strictly *greater than* the volume of the 3D projection of the block it encapsulates.
3. While also having a larger volume, the smaller block should also fit inside of the larger block.
4. Rotations of the Block create different 3D projections of the same object, just as rotations of a 3D block have different base areas.
5. When you rotate the block, the values for $w(i), x(i), y(i), z(i)$ will correspond to different dimensions, depending on the rotation, i.e. what was initially $w(i)$ may become the value for $x(i)$ depending on the rotation.
6. Your goal is to maximize the dimension not included in the 3D projection, which will be $w(i)$ *after* the rotation.

2 Recursion

2.1 Notation

The next important step to Dynamic Programming is to choose a system of notation that will provide the necessary means to get us to a solution. We need to be able to refer to the four different dimensions for the i^{th} block, and access them as such:

- We will refer to the dimensions as $w(i), x(i), y(i), z(i)$.
- The Max Sum w with block i at the center will be MSW(i)

[1] This problem seems very daunting at first, however, once you get an understanding of the 3D version, the 4D instance is a fairly straightforward extension of the 3D version of the problem. You should consider looking at the 3D problem to gain an intuition for the goal and the process, and then it will be easier to extend that understanding to the 4D version. Essentially you stack 3D blocks on top of one another to maximize the height of the tower, with the base of the block on top being strictly smaller than the base of the block on bottom.

2.2 Recursion

We are trying to fit 3D projections of 4D blocks into each other while maximizing the sum of the 4th Dimension of all blocks contained. With our list of the $4n$ rotations for the blocks, we calculate the volume of the 3D projection for each, $x * y * z$ and we then sort it in decreasing order of volume. With this list of volumes for the 3D projections, keeping the corresponding values for $w(i)$ this problem becomes an instance of the longest increasing sub-sequence problem, where we are trying to find the sequence of 3D projections that fit within each other that maximizes the sum of the dimension $w(i)$. Finding the recurrence relation to this can be fairly straightforward we just want to ensure that the recurrence relation satisfies the constraints the problem has.

We now look for the a sub problem structure that we can use to develop a recurrence relation. if we want to calculate the sum of w for our current tower with i on top, we need to need to calculate the sum of w for the tower on which i rests, we do this by finding the maximum w of a tower with volumes greater than that of i, since the list of volumes is in descending order these are in the range $0 < j < i$. We then add the value of $w(i)$ to the max value of $MSW(j)$. We repeat this process until we get to a point where there is no such value i exists, in other words i is the base of the tower in the 3D problem or the outermost block (containing all other blocks) in this problem.

$$\text{MSW}(i) = \begin{cases} max_{0<j<i}(\text{MSW}(j)) + w(i) & [x(j) > x(i)]\&[y(j) > y(i)]\&[z(j) > z(i)] \\ w(i) & \text{If no such } j \text{ exists.} \end{cases}$$

3 Primer Questions

1. What does it mean if any of the conditions for the main call are violated? What is it's parallel in the 3D problem, what does it mean in the context of this problem?

2. What do the base cases represent?

3.