Algorithms
DP homework 1

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Problem #1: Subset Sum Problem

- Given an array of N positive integers, and a value target, determine if there is a subset of the given numbers with sum equal to given target.
  - N <= 500 and Target <= 2000
  - Value[i] <= 10^8
- Input ⇒ Output
  - {3, 12, 4, 12, 5, 2}, sum = 9 ⇒ True from 4+5
  - {3, 40, 4, 12, 5, 2}, sum = 30 ⇒ False
- C++: bool subset_sum(cons vector<int> &values, int target)
- Python: def subset_sum(values, target)
- The function returns true if such subset exists
- **Hint**: Think like a knapsack (pick or leave)

Problem #2: LeetCode 416 - Partition Equal Subset Sum

- Given a non-empty array containing N positive integers, find if the array can be **partitioned into two subsets** such that the sum of elements in both subsets is equal.
  - N <= 200 and values[i] <= 100
- Input ⇒ Output
  - [1,5,11,5] ⇒ True   [from [1, 5, 5] and [11]]
  - 1,2,3,5] ⇒ False
- **Hint**: Find a **simple** observation to **reduce** this problem to the previous one
  - Sometimes it is impossible/hard to apply DP on a problem, but on a **reduced** problem

Problem #3: [LeetCode 1691](#) - Maximum Height by Stacking Cuboids

Given `n` `cuboids` where the dimensions of the `i`th cuboid is `cuboids[i]` = `[width`$_i$`, length`$_i$`, height`$_i$`]` (**0-indexed**). Choose a **subset** of `cuboids` and place them on each other.

You can place cuboid `i` on cuboid `j` if `width`$_i$ `<= width`$_j$ and `length`$_i$ `<= length`$_j$ and `height`$_i$ `<= height`$_j$. You can rearrange any cuboid's dimensions by rotating it to put it on another cuboid.

Return the **maximum height** of the stacked `cuboids`.

- 1 <= cuboids. <= 100
- 1 <= width, length, height <= 100
- **Hint**: Find observation to use LIS
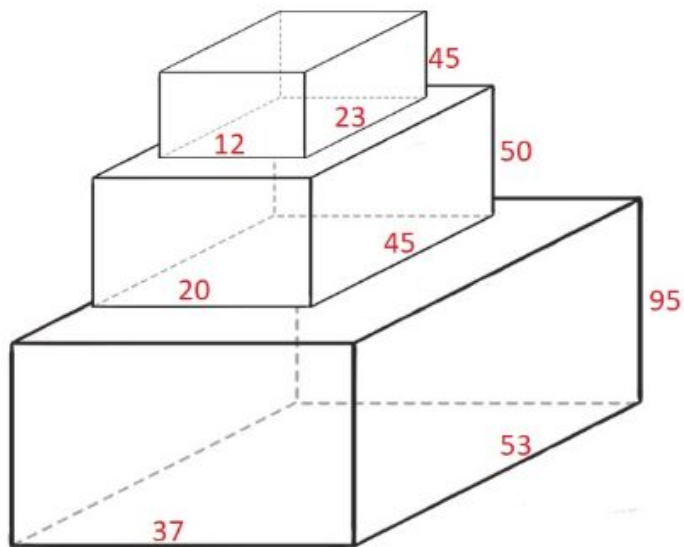
Input: cuboids = [[50,45,20],[95,37,53],[45,23,12]]
Output: 190
Explanation:
Cuboid 1 is placed on the bottom with the 53x37 side facing down with height 95.
Cuboid 0 is placed next with the 45x20 side facing down with height 50.
Cuboid 2 is placed next with the 23x12 side facing down with height 45.
The total height is 95 + 50 + 45 = 190.

Example 2:

```
Input: cuboids = [[38,25,45],[76,35,3]]
Output: 76
Explanation:
You can't place any of the cuboids on the other.
We choose cuboid 1 and rotate it so that the 35x3 side is facing down and its
height is 76.
```

Example 3:

```
Input: cuboids = [[7,11,17],[7,17,11],[11,7,17],[11,17,7],[17,7,11],[17,11,7]]
Output: 102
Explanation:
After rearranging the cuboids, you can see that all cuboids have the same
dimension.
You can place the 11x7 side down on all cuboids so their heights are 17.
The maximum height of stacked cuboids is 6 * 17 = 102.
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."