

# Problem 3259: Maximum Energy Boost From Two Drinks

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 49.75%

**Paid Only:** No

**Tags:** Array, Dynamic Programming

## Problem Description

You are given two integer arrays `energyDrinkA` and `energyDrinkB` of the same length `n` by a futuristic sports scientist. These arrays represent the energy boosts per hour provided by two different energy drinks, A and B, respectively.

You want to \_maximize\_ your total energy boost by drinking one energy drink \_per hour\_. However, if you want to switch from consuming one energy drink to the other, you need to wait for \_one hour\_ to cleanse your system (meaning you won't get any energy boost in that hour).

Return the \*\*maximum\*\* total energy boost you can gain in the next `n` hours.

\*\*Note\*\* that you can start consuming \_either\_ of the two energy drinks.

\*\*Example 1:\*\*

\*\*Input:\*\* energyDrinkA = [1,3,1], energyDrinkB = [3,1,1]

\*\*Output:\*\* 5

\*\*Explanation:\*\*

To gain an energy boost of 5, drink only the energy drink A (or only B).

\*\*Example 2:\*\*

**\*\*Input:\*\*** energyDrinkA = [4,1,1], energyDrinkB = [1,1,3]

**\*\*Output:\*\*** 7

**\*\*Explanation:\*\***

To gain an energy boost of 7:

- \* Drink the energy drink A for the first hour.
- \* Switch to the energy drink B and we lose the energy boost of the second hour.
- \* Gain the energy boost of the drink B in the third hour.

**\*\*Constraints:\*\***

\* `n == energyDrinkA.length == energyDrinkB.length` \* `3 <= n <= 105` \* `1 <= energyDrinkA[i], energyDrinkB[i] <= 105`

## Code Snippets

### C++:

```
class Solution {
public:
    long long maxEnergyBoost(vector<int>& energyDrinkA, vector<int>&
    energyDrinkB) {
        }
};
```

### Java:

```
class Solution {
    public long maxEnergyBoost(int[] energyDrinkA, int[] energyDrinkB) {
        }
}
```

### Python3:

```
class Solution:
    def maxEnergyBoost(self, energyDrinkA: List[int], energyDrinkB: List[int]) ->
```

