**ARRAY**

Table of Contents

[LEVEL 1: **EASY** 2](#_Toc171402490)

[1. Average Waiting Time 2](#_Toc171402491)

[2. Name 2](#_Toc171402492)

[LEVEL 2: **Medium** 3](#_Toc171402493)

[LEVEL 3: **Difficult** 4](#_Toc171402494)

[**SOLUTIONS:** 5](#_Toc171402495)

[LEVEL 1**:** 5](#_Toc171402496)

# LEVEL 1: **EASY**

### Average Waiting Time

Link: <https://leetcode.com/problems/average-waiting-time/description/>

### Name

Link: <link>

# LEVEL 2: **Medium**

# LEVEL 3: **Difficult**

# **SOLUTIONS:**

## **LEVEL 1:**

1. Average Waiting Time

This problem can be solved by simulating the sequence of events efficiently. Instead of a naive simulation of each time step, which would be too slow, we process each customer as they arrive. For simulation problems like this, the key is to track and correctly update the variables involved in the simulation.

class Solution:

    def averageWaitingTime(self, customers: List[List[int]]) -> float:

        available\_at = 0

        total\_wait = 0

        for arrival, t in customers:

            available\_at = max(available\_at, arrival) + t

            total\_wait += available\_at - arrival

        return total\_wait / len(customers)