Problem 1094: Car Pooling

Problem Information

Difficulty: Medium
Acceptance Rate: 56.14%
Paid Only: No
Tags: Array, Sorting, Heap (Priority Queue), Simulation, Prefix Sum

Problem Description

There is a car with `capacity` empty seats. The vehicle only drives east (i.e., it cannot turn around and drive west).

You are given the integer `capacity` and an array `trips` where `trips[i] = [numPassengersi, fromi, toi]` indicates that the `ith` trip has `numPassengersi` passengers and the locations to pick them up and drop them off are `fromi` and `toi` respectively. The locations are given as the number of kilometers due east from the car's initial location.

Return `true` _if it is possible to pick up and drop off all passengers for all the given trips, or_`false` _otherwise_.

Example 1:

Input: trips = [[2,1,5],[3,3,7]], capacity = 4 **Output:** false

Example 2:

Input: trips = [[2,1,5],[3,3,7]], capacity = 5 **Output:** true

Constraints:

* `1 <= trips.length <= 1000` * `trips[i].length == 3` * `1 <= numPassengersi <= 100` * `0 <= fromi < toi <= 1000` * `1 <= capacity <= 105`

Code Snippets

C++:

```
class Solution {
public:
bool carPooling(vector<vector<int>>& trips, int capacity) {

}
};
```

Java:

```
class Solution {
public boolean carPooling(int[][] trips, int capacity) {

}
}
```

Python3:

```
class Solution:
def carPooling(self, trips: List[List[int]], capacity: int) -> bool:
```