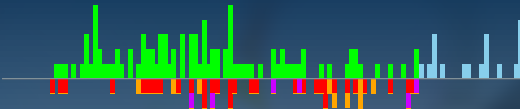


# J: Juggling Keys

Markus Himmel

## Problem

Given  $n$  people sharing an apartment with  $k$  keys going on  $q$  trips, find who needs to take a key such that no one arrives to an empty apartment without a key.

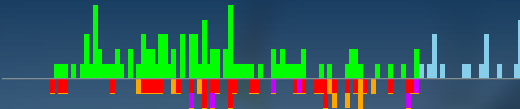


## Problem

Given  $n$  people sharing an apartment with  $k$  keys going on  $q$  trips, find who needs to take a key such that no one arrives to an empty apartment without a key.

## Observation

A trip only needs a key if it ends when the apartment is empty.

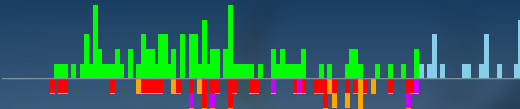


## Problem

Given  $n$  people sharing an apartment with  $k$  keys going on  $q$  trips, find who needs to take a key such that no one arrives to an empty apartment without a key.

## Solution

- Simulate while keeping track of the number of people in the apartment.
  - Loop through a sorted array of all arrivals and departures.
  - If a person arrives to an empty apartment, they should have taken a key.

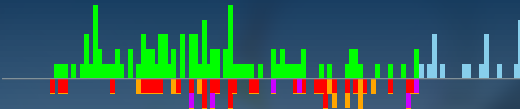


## Problem

Given  $n$  people sharing an apartment with  $k$  keys going on  $q$  trips, find who needs to take a key such that no one arrives to an empty apartment without a key.

## Solution

- Simulate while keeping track of the number of people in the apartment.
  - Loop through a sorted array of all arrivals and departures.
  - If a person arrives to an empty apartment, they should have taken a key.
- Do a second simulation while keeping track of the number of keys in the apartment.
  - If this ever becomes negative, output “impossible”.

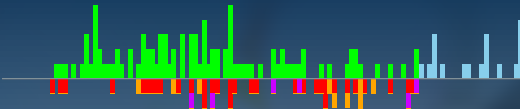


## Problem

Given  $n$  people sharing an apartment with  $k$  keys going on  $q$  trips, find who needs to take a key such that no one arrives to an empty apartment without a key.

## Solution

- Simulate while keeping track of the number of people in the apartment.
  - Loop through a sorted array of all arrivals and departures.
  - If a person arrives to an empty apartment, they should have taken a key.
- Do a second simulation while keeping track of the number of keys in the apartment.
  - If this ever becomes negative, output “impossible”.
- Runtime:  $\mathcal{O}(n \log n)$ .



## Problem

Given  $n$  people sharing an apartment with  $k$  keys going on  $q$  trips, find who needs to take a key such that no one arrives to an empty apartment without a key.

## Solution

- Simulate while keeping track of the number of people in the apartment.
  - Loop through a sorted array of all arrivals and departures.
  - If a person arrives to an empty apartment, they should have taken a key.
- Do a second simulation while keeping track of the number of keys in the apartment.
  - If this ever becomes negative, output “impossible”.
- Runtime:  $\mathcal{O}(n \log n)$ .