

# Documentation for the "Minimum Time Difference"

## 1. Problem Statement

Given a list of time points in 24-hour "HH:MM" format, return the minimum difference in minutes between any two time-points in the list.

Constraints:

- $2 \leq \text{timePoints.length} \leq 2 * 10^4$
- Each time is valid and formatted as "HH:MM"

## 2. Intuition

The smallest time difference can only be found by converting all time strings to integers representing minutes since midnight. After sorting them, we can find the smallest absolute difference between adjacent elements, and also consider the wrap-around case (last and first).

## 3. Key Observations

- 1 day = 1440 minutes. If the number of time points  $> 1440$ , a duplicate must exist  $\rightarrow$  answer is 0.
- Time is circular (i.e., 23:59 to 00:00 is only 1 minute apart).
- After sorting times (in minutes), only adjacent comparisons are needed for minimum difference.

## 4. Approach

- Convert each "HH:MM" string into total minutes ( $h * 60 + m$ ).
- Sort the list of minutes.
- Calculate the difference between each pair of adjacent time points.
- Compute the circular difference between the last and first elements.
- Return the minimum difference.

## 5. ⚠ Edge Cases

- Duplicate time points  $\rightarrow$  return 0
- Time difference across midnight  $\rightarrow$  must compute  $1440 - (\text{last} - \text{first})$
- Input list of max size  $\rightarrow$  should not exceed memory or runtime limits

## 6. 📊 Complexity Analysis

### □ Time Complexity

- $O(N \log N)$ : due to sorting of  $N$  time points

### □ Space Complexity

- $O(N)$ : for storing converted time points in minutes

## 7. 🔄 Alternative Approaches

### a) Boolean Bucket Array (Optimized)

- Create a `boolean[1440]` to mark seen minutes.
- If a duplicate is found while inserting  $\rightarrow$  return 0
- Then iterate through array to find the minimum difference.
- Time:  $O(N)$
- Space:  $O(1)$  (constant size array)

## 8. ✓ Test Cases

Input	Output	Explanation
<code>["23:59","00:00"]</code>	1	Circular case across midnight
<code>["00:00","23:59","00:00"]</code>	0	Duplicate present
<code>["12:00", "12:01", "12:03"]</code>	1	Min difference between 12:00 and 12:01
<code>["01:01", "02:01", "03:00"]</code>	59	Closest is 02:01–03:00
<code>["00:00", "12:00", "23:59"]</code>	719	Closest is 00:00–23:59

## 9. □ Final Thoughts

- Always sort converted times and check circular difference.
- For ultra-optimized versions, use a boolean array for fixed 1440-minute space.
- Handle duplicates early to avoid unnecessary computation.