Problem Description

Next Closest Time

Given a `time` represented in the format `"HH:MM"`, form the next closest time by reusing the current digits. There is no limit on how many times a digit can be reused.

You may assume the given input string is always valid. For example, `"01:34"`, `"12:09"` are all valid. `"1:34"`, `"12:9"` are all invalid.

Example 1:

```
Input: time = "19:34"
Output: "19:39"
Explanation: The next closest time choosing from digits 1, 9, 3, 4, is 19:39, which occurs 5 minutes la
It is not 19:33, because this occurs 23 hours and 59 minutes later.
```

Example 2:

```
Input: time = "23:59"
Output: "22:22"
Explanation: The next closest time choosing from digits 2, 3, 5, 9, is 22:22.
It may be assumed that the returned time is next day's time since it is smaller than the input time num
```

Constraints:

- `time.length == 5`
- `time` is a valid time in the form `"HH:MM"`.
- `0 <= HH < 24`
- `0 <= MM < 60`

Analysis

01:34

H2	H1	M2	M1
0	1	3	4

{0,1,3,4}
M1 == 4
M2 == 3 Return 01:43

{1, 3, 4, 9}
19:34 → 19:39
{1,3,9}
19:39 → 11:11
{2,3,5,9}
23:59 → 22:22

H2	H1	:	M2	M1
1	9		3	4

0 <= H2 <= 1 and 0 <= H1 <= 9
H2 = 2 and 0 <= H1 <= 3

0 <= M2 <= 5 and 0 <= M1 <= 9
0 <= M2 <= 5 and 0 <= M1 <= 9

- `0 <= HH < 24`
- `0 <= MM < 60`

Algorithm

Read in all distinct numbers: S, rank it.
ab:cd
For M1, test if there exists e in S such that d < e <= 9. If yes, return ab:ce.
For M2, test if there exists e in S such that c < e <= 5. If yes, return ab:ed', where d' = min(S).
For H1,
If H2 =2, test if there exists e in S such that b < e <= 3. If yes, return ae:d'd', where d' = min(S).
Else, test if there exists e in S such that b < e <= 9. If yes, return ae:d'd', where d' = min(S).
Else, find e = min(S). Return worstCase(S).
For H2, test if there exists e in S such that a < e <= 2, If yes, return ed':d'd', where d' = min(S).
Else, call worstCase(S).

Method: worstCase

This method takes in S and return ee:ee where e = min(S).