Kick Start 2022 - Coding Practice with Kick Start Session #3

Analysis: Record Breaker

View problem and solution walkthrough video

To check whether i is a record breaking day, we make two checks:

- The number of visitors on day i ($\mathbf{V_i}$) must be greater than the number of visitors for any previous days.
- Either it is the last day OR the number of visitors on the day i are more than the day $i+1$.

The second one is easy to check in constant time. Checking the first one is the difficult part of this problem. Here's how we would do that:

Test Set 1

For each element j such that $(1 \leq j < i)$, check that the number of visitors on day j are less than number of visitors on day i. In other words $(\mathbf{V_j} < \mathbf{V_i})$. Hence, for each day we would compare it with all the previous days and it would take $O(\mathbf{N})$ time. Therefore, for $\mathbf{N}$ days, the time complexity of this solution would be $O(\mathbf{N^2})$.

Test Set 2

However, a solution that takes $O(\mathbf{N^2})$ time is not fast enough for Test Set 2, so we need a faster approach. Instead of comparing the number of visitors of day i against *all* the previous days one by one, we can compare the number of visitors of day i against the *greatest number of visitors* from all previous days. That reduces our processing time for each day from O($\mathbf{N}$) to O(1). Therefore, for $\mathbf{N}$ days, the time complexity of this solution would be O($\mathbf{N}$), which is sufficiently fast for both Test Set 1 and Test Set 2.

Sample Code (C++)

```
int countRecordBreakingDays(vector<int> visitors) {
  int recordBreaksCount = 0;
  int previousRecord = 0;
  for(int i = 0; i < checkpoints.size(); i++) {
     bool greaterThanPreviousDays = i == 0 || visitors[i] > previousRecord;
     bool greaterThanFollowingDay = i == checkpoints.size()-1 || visitors[i] > visitors[i+1];
     if(greaterThanPreviousDays && greaterThanFollowingDay) {
        recordBreaksCount++;
     }
     previousRecord = max(previousRecord, visitors[i]);
  }
  return recordBreaksCount;
}
```