Kick Start 2019 - Round D

Analysis: Latest Guests

Test set 1 (Visible)

For this test set, we simply simulate the entire affair. For each second, we update the positions of all the guests and will add them to the list of latest guests list of the consulate that they visit. We also keep track of the minute when the list for each consulate was updated so that if a guest comes at a later time, we discard the old list and put this guest (and possibly others) to a new list. Finally after the **M**th minute, we iterate through all these lists and find for each guest, how many lists they were a part of and output the answer. This takes a total of O(**G**x(**M**+**N**)) time which is sufficient for this test set.

Test set 2 (Hidden)

Let's simplify the problem by assuming all the guests travel clockwise. We can get the final position of each guest after **M** minutes by simply taking their current position, adding **M** to it and finding the modulo of the result with **N**. We categorise the guests into groups based upon their final positions. All guests having the same final positions are in the same group and guests having different final positions are in different groups. We maintain a mapping from guests to groups. We are going to find for each group, how many consulates remember it.

Now notice that for any consulate only the group whose final position is on or just after it in a clockwise order can be the last visiting group. We can get this by doing a binary search on the sorted list of final positions of the groups or using a sliding window. Note that there can be at most one group which will be remembered by a consulate.

If this group did not visit this consulate then no one has visited it. To find this we find the time when the group would have visited this consulate last. So we take the difference of their final positions and the position of the consulate and subtract this number from **M**. If the resultant is negative that means no one visited this consulate. Otherwise, for this consulate, we make a note of this group and the time when they visited.

We handle anti-clockwise guests similarly and get for each consulate the last anti-clockwise guest group to visit it and the time when the group visited.

Now we iterate through the consulates and see which was the last group to visit it (we include both the clockwise and the anti-clockwise groups). We then increment the number of consulates that remembers this group. Finally, we iterate through the guests and see which group they belong to and output the number of consulates that remember this group. This solution has a run-time of O(**N+G**) if implemented using the sliding window technique or a run-time of O(**G** + **N**log(**N**)) if we use binary-search instead.