Code Jam to I/O for Women 2014 - Code Jam to I/O for Women

Analysis: Seating Chart

Seating Chart: Analysis

Before assigning any particular people to particular tables, we can figure out how many people must sit at each table: M = **N** mod **K** of the tables will be "more full" and will have ceil(**N** / **K**) people each, and the other L tables will be "less full" and will have floor(**N** / **K**) people each.

Small dataset

In the Small dataset, there are at most 8 people, and at most 8 tables. These numbers are small enough that we can try a brute force strategy; here is one example. Number the people from 1 to **N**, and generate each of the **N**! permutations of the list [1, ..., **N**]. Then, for each permutation, proceed left to right through the permutation, filling up the first table in clockwise order with the appropriate number of people, then the second table, and so on. Turn each of these assignments into a sorted list of pairs of people who are sitting next to each other, with the numbers in each pair themselves sorted in ascending order. Maintain a set of all such lists found, discarding any duplicates, and then return the size of that set.

Large dataset

In the Large dataset, **N** can be as large as 20, so we need to use combinatorics. As in the Small dataset, the tricky part is to avoid counting the same arrangement more than once.

First, let's consider how we assign people to tables. If we have P people who are not yet assigned, and the table requires Q people, there are (P choose Q) ways of selecting those people.

Then, how many ways are there to put those Q people around the table? First, let's dispense with some small cases: for 1 or 2 people, there is only one way. Otherwise, without loss of generality, let's put the first person in our chosen set at the "top" of the table. Then, for each of the (Q-1)! possible orders of the remaining people, we can try putting them clockwise around the table, starting from the spot that is directly clockwise from our "top" person. Notice that for any one of these orders, using the reverse of that order creates an equivalent arrangement, so there are really only (Q-1)! / 2 possibilities.

So, we can start by saying there is 1 possible arrangement, and then, for each table, multiply that number by (P choose Q) and (Q-1)! / 2, to reflect populating that table. The value of P will diminish as we seat more and more people, and the value of Q will depend on whether we are looking at one of our "more full" or "less full" tables. It does not matter what order we process the tables in.

However, after we have done this, we still need to correct for some overcounting. For any particular arrangement, we have a huge number of duplicates. Specifically, we could have enumerated the M "more full" tables in the arrangement in any one of M! possible orders, and the L "less full" tables in any one of L! possible orders. Dividing our total by M! and L! eliminates this redundancy and yields the correct answer.