Group HW #01

Abby Omer, Adi Sudhaker, Antoinette Tan, Dasha Chadiuk

November 1, 2021

- **ALL DID ALL**
- **Resources used:** We referenced the class textbook and the answers to the second in class work.
- **Group goal this week:** To collaborate in a way that is efficient.

Team Contract

All members (Abby, Adi, Antoinette, and Dasha) will meet on Saturday 9am each week to work on group problems. To prepare for these meetings, everyone is expected to attempt each homework problem before the meeting. In addition, one person each week will write up the report to be turned in. The Latex document will be sent to everyone in the group the Sunday morning before the homework is due so that each group member can look over the report before it is submitted. This task of writing up the report will be assigned to people on a rotation basis for each group homework.

Problem 1

Problem: How many subsets of a set with 100 elements have more than one element?

Let S be a finite set with 100 elements. To generate a subset of S, each element is either in the subset or not in the subset. Therefore, by product rule, there are 2^{100} subsets in total. However, we must remove the subsets with one or less elements. There are 100 subsets with one element, and 1 subset with no elements. Therefore, the number of subsets of S with more than one element is $2^{100} - 100 - 1$

Problem 2

Problem: How many bit strings of length seven either begin with two 0s or end with three 1s?

For this problem, we used the principle of inclusion-exclusion. The inclusion-exclusion principle applies when two tasks can be done at the same time. When this occurs, we must add the number of ways to do each of the two tasks and then subtract the number of ways to do both tasks. This principle can also be explained with the following:

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|,$$

where A_1 contains the elements which fulfill task T_1 and A_2 contains the elements which fulfill task T_2, $||$ denotes cardinality, $\cup$ denotes set union, and $\cap$ denotes set intersection.

In this case, our first task (T_1) is constructing a bit string of length seven beginning with two 0s, and our second task (T_2) is constructing a bit string of length seven ending with three 1s.

The first task, constructing a bit string of length seven beginning with two 0s can be done in $2^5 = 32$ ways. The second task, constructing a bit string of length seven ending with three 1s can be done in $2^4 = 16$ ways. Both tasks, constructing a bit string of length seven that begins with two 0s and ends with three 1s, can be done in $2^2 = 4$ ways. Consequently, the number of bit strings of length seven that begin with two 0s or end in three 1s is equal to 32 + 16 - 4 = 44.

Problem 3

Problem: How many bit strings of length 10 contain either five consecutive 0s or five consecutive 1s?

To solve this problem, we first counted the number of bit strings of length 10 containing five consecutive 0s. To think about this problem, we thought about each of the positions that the 0s could be, and how many options that would leave us with. For the first position, 00000_____, we have one choice for each of the slots with a 0, and two choices in the remaining 5 slots. This leads us with 2^5 strings by the product rule. Next, for the second position, _00000____, we know that the first bit must be a 1, otherwise we would be over-counting a case. Therefore, there is one choice for the first 6 slots, and two choices in the remaining 4 slots. This leads us with 2^4 strings by the product rule. For the third, fourth, and sixth positions, we can follow the same logic as applied above (placing a 1 in the slot before the consecutive 0s and having 4 remaining slots). Therefore, the number of bit strings with five consecutive 0s is $2^5 + 2^4 * 5 = 112$.

Next, we counted the number of bit strings of length 10 containing five consecutive 1s. Since the number of consecutive 1s is the same as the number of consecutive 0s, we can use the same exact logic we used for the five consecutive 0s and apply it here. Therefore there are also 112 cases for the number of bit strings of length 10 containing five consecutive 1s.

Lastly, we accounted for any overlap in the cases. Specifically, we must account for the cases where there are both five consecutive 1s and 5 consecutive 0s in the same bit string of length 10. Because there are only 10 digits, the only cases for overlap are when there are only the five 0s and five 1s. Therefore, the two cases for overlap are 0000011111 and 1111100000. Using inclusion-exclusion, we can get a final answer of 112 + 112 - 2 = 222.

Problem 4

Problem: How many integers in the range 1000-9999 (inclusive) do NOT have any repeated digits?

To approach this problem, we can think about each digit as a slot. For the first slot, the digits 1 through 9 can be chosen. This gives us nine choices for the first slot. For the second slot, the digits 0 through 9 can be chosen as long as it is not the same number as the first slot. Therefore, the second slot has nine choices. For the third slot, the digits 0 through 9 can be chosen as long as it is not the same number as the first or second slots. Therefore, the third slot has eight choices. For the fourth slot, the digits 0 through 9 can be chosen as long as it is not the same number as the first, second, or third slots. Therefore, the fourth slot has seven choices. Using the product rule, this leaves us with $9 * 9 * 8 * 7 = 4536$ choices where the integers do not have any repeated digits.

Problem 5

Problem: If you select 5 integers at random from the set N =1, 2, ..., 8, prove that some 2 of those numbers selected must sum to exactly 9.

Within the set, N, there are four distinct pairs of numbers in which the pair sums to 9. We can think about the 4 pairs as holes, and the 5 selected integers as pigeons. By the pigeon-hole principle, at least two of the five selected integers will be in a pair together as there are more pigeons than holes.

Problem 6

Problem: NOTE: For each part, write a complete argument without making any reference to any other part.

Part A): Explain in detail why $\binom{6}{0} + \binom{6}{1} + \binom{6}{2}$ counts the number of bitstrings of length 6 that have at most two 1s.

Part B): Explain in detail why $2^6 - \binom{6}{3} - \binom{6}{4} - \binom{6}{6}$ also counts the number of bitstrings of length 6 that have at most two 1s.

Part C): Explain in detail why $\binom{6}{4} + \binom{6}{5} + \binom{6}{6}$ also counts the same thing.

To answer this problem, we first need to define the notation above and how they relate to combinatorics: For a number of r-combinations of a set with n elements, where n is a nonnegative integer and r is an integer with $0 \leq r \leq n$, $\binom{n}{r} = \frac{n!}{r!(n-r)!}$. This is because the r-permutations of a set can be obtained by forming the $\binom{n}{r}$ r-combinations of the set, and then ordering the elements in each r-combination, which can be done in P(r, r) ways. Therefore,

$$P(n, r) = \binom{n}{r} * P(r, r),$$

which implies that

$$\binom{n}{r} = \frac{P(n, r)}{P(r, r)} = \frac{n!/(n-r)!}{r!/(r-r)!} = \frac{n!}{r!(n-r)!}$$

Part A): $\binom{6}{0}$ is used to describe all length 6 bit strings with zero 1s, $\binom{6}{1}$ is used to describe all length 6 bit strings with one 1s, and $\binom{6}{2}$ is used to describe all length 6 bit strings with two 1s. By counting all these cases, we can account for the number of bitstrings of length 6 that have at most two 1s.

Part B): 2^6 represents all of the possible bitstrings of length 6 as defined by the product rule. $\binom{6}{3}$ can be used to describe all length 6 bit strings with three 1s, $\binom{6}{4}$ can be used to describe all length 6 bit strings with four 1s, $\binom{6}{5}$ can be used to describe all length 6 bit strings with five 1s, and $\binom{6}{6}$ can be used to describe all length 6 bit strings with six 1s. By taking all of the possible bitstrings, and subtracting the cases where there are more than two 1s, we are then left with all the bitstrings where there are at most two 1s.

Part C): $\binom{n}{r} = \binom{n}{n-r}$. Therefore, to count the number of bitstrings with two 1s, we can use $\binom{6}{4}$, to count the number of bitstrings with one 1, we can use $\binom{6}{5}$, and to count the number of bitstrings with no 1s, we can use $\binom{6}{6}$. By counting all these cases, we can account for the number of bitstrings of length 6 that have at most two 1s.

Problem 7

Problem: Suppose there is a research conference reception with at least two people. Prove that there must be (at least) two people at this reception who have collaborated with the same number of other people at this reception. (Assume "collaboration" is symmetric so that two given people either both have collaborated with each other or neither has collaborated with the other.)

Say that there are N people in total. Each person can collaborate with a maximum N-1 people because no one person can collaborate by themselves. This means that each person has collaborated with 0 to (N-1) other people. Let us consider two cases.

Case 1: There exists a person with N-1 collaborations.

This means that they have collaborated with every other person and that every other person has collaborated with at least them. Therefore it is impossible for a person at the conference to then have 0 collaborations and the range of number of collaborations becomes 1 to (N-1). With N-1 numbers in this range and N people at the conference, there must be two people who have the same number of collaborations by the pigeonhole principle.

Case 2: There exists no people with N-1 collaborations.

The range for the number of collaborations becomes 0 to (N-2). With N-1 numbers in this range and N people at the conference, there must be two people who have the same number of collaborations by the pigeonhole principle.

Since there must either be someone with N-1 collaborations or nobody with N-1 collaborations and in both cases there must be two people who have the same number of collaborations, we have proven this problem.

Team Report

Our team worked well together. Our goal this week was to collaborate in a way that is efficient. Because everyone tried the problems before we met, the meeting was able to go smoothly and efficiently. We went through the problems as a group, taking turns explaining each question, making sure that everyone was on the same page. Then, for the final Latex report, one person wrote it up while everyone else was able to proof-read and suggest edits.