Counting — CSCI 104 — Devin C. Martin

① Unique Subsets of 5 Letters

Problem: Count the # of unique subsets of 5 letters from the 7-letter word 'unusual'

Solution: Since all letters in 'unusual' are distinct, it's a simple combination problem

$$\binom{7}{5} = \frac{7!}{5! * (7-5)}$$

Answer: 21 unique Subsets

Different Strings of 5 Letters.

Problem: Count the # of diff strings that can be made from 5 of the 7 letters in 'unusual'.

Solution: This is a permutation problem since the order of letters matters.

$$P(7,5) = \frac{7!}{(7-5)!}$$

Answer: 2,520 diff strings

② 5-Card Hand w/ 2 Pairs from a Standard Deck

Problem: Calculate the # of ways to form a 5-card hand w/ 2 pairs using a standard deck of playing cards

Solution: Choose two ranks for the pairs, two suits for each pair, and a fifth card of a diff rank and suit

$$\binom{13}{2} * \binom{4}{2}^2 * 11 * 4$$

Answer: 123,552 ways

③ Distribution of Songs Among Couples

Problem: 16 total songs, 7 total couples, and a restriction of 1 couple can recieve at most 1 song

Solution Approach: This can be approached by partitioning the 16 songs among the 7 couples, taking into consideration the restriction of one couple

Case 1: The Restricted Couple Recieves 1 Song

→ We assign 1 song to the couple w/ the restriction

→ Now, we have 15 songs left to distribute among the remaining 6 couples

→ This is a problem of partioning 15 items into 6 parts, where each part can take any value from

0 to 15

→ If we ignore the condition that each couple must recieve at least one song, each of the 15 songs can be given to any of the 6 couples, leading to 6^{15} possible distributions

Case 2: The Restricted Couple Recieves 0 songs

→ the couple w/ the restriction recieves no songs

→ we have 16 songs to distribute among the 6 other couples

→ this is a problem of partioning 16 items into 6 parts, where each part can take any value from 0 to 16

→ Similarly, ignoring the at-least-one-song condition, there are 6^{16} possible distributions

Total Approximate Count:

→ The total # of approx. distributions is the sum of the distributions in both cases

→ This approach overestimates the actual # bcuz it includes distributions where some couples might not recieve any songs

Calculation: Let's solve for the total approx. possibilities for both cases:

→ Total Approx Possibilities for both cases:

→ Total Approx Possibilities $= 6^{15} + 6^{16}$

→ This will provide us w/ a rough estime of the #

of distributions. Now, let's perform the calculation

```
# Calculating the approx total possibilities for both cases
approx_possibilities_case_1 = 6**15 #15 songs, 6 couples
approx_possibilities_case_2 = 6**16 #16 songs, 6 couples
#Total Approx Possibilities
total_approx_possibilities = approx_possibilities_case_2
                              + approx_possibilities_case_2
total_approx_possibilities
```

The approx total # of ways to distribute the 16 songs among the 7 couples, considering the constraint on one couple, is

approx = 3,291,294,892,032

The estimate includes scenarios where in some cases some couples might not receive any songs.

4) Binary Search Trees from Given Nodes

Problem: 12 total nodes, w/ distinct values btwn 1 & 12, and specific conditions including the root having a value of 3 and the right child of the root having a value of 9.

Solution Approach: This problem involves counting the # of distinct BSTs that can be formed under these conditions. The key property of a BST is that, for any node, all values in its left subtree are smaller, and all values in its right subtree are larger.

1. Distribution of Nodes:
   - → Nodes less than 3 (1 & 2) must be in the left subtree of the root
   - → Nodes btwn 3 & 9 (4, 5, 6, 7, 8) can be in the right subtree of the root but the left subtree of node 9
   - → Nodes greater than 9 (10, 11, 12) must be in the right subtree of node 9

2. Coming BSTs for Each Subtree:
   - → the # of BSTs that can be formed w/ a set of nodes is given by the Catalan # C_n, where n is the # of nodes
   - → for n nodes, the Catalan # is $C_n = \frac{1}{n+1}\binom{2n}{n}$

3. Calculating Total BSTs:
   - → Left Subtree of Root (3): 2 nodes (1, 2)
     - → # of BSTs: C_2

→ Right Subtree of Root but Left of Node(9):
5 nodes (4, 5, 6, 7, 8)
→ # of BSTs: C_5

→ Right Subtree of Node(9): 3 nodes (10, 11, 12)
→ # of BSTs: C_3

→ The total # of BSTs is the product of these counts.

Calculation: Let's calculate the total # of BSTs using these steps, by solving for the Catalan #'s for each subtree & then multiply them together

```
# Calculating the Catalan nums for each subtree
catalan_2 = catalan(2) #Catalan num for 2 nodes (left of root)
catalan_5 = catalan(5) # Catalan num for 5 nodes (right of root,
                                                   left of 9)
catalan_3 = catalan(3) # catalan num for 3 nodes (right of 9)
# Total num of BSTs is the product of these catalan nums
total_bsts = catalan_2 * catalan_5 * catalan_3
total.evalf() # evaluate to get a numerical value
```

The total # of distinct BSTs that can be formed w/ 12 nodes, given that the root is 3 & its right child is 9 = 420

This calculation uses the Catalan #'s, which are central in counting the # of BSTs possible w/ a given # of nodes. Each subtree's configuration contributes to the total count, & the product of these counts gives the final answer

5 Combinations of Patients Served by Nurses

Problem: There are total patients consisting of 10 friends arriving for a COVID vax, 4 total nurses, but 2 may be on break, and lastly a constraint of each working nurse serving at least one person.

Solution Approach: This problem can be approached as a partition problem in combinatorics, where we need to partition 10 patients among up to 4 nurses. We have to consider diff scenarios based on the # of nurses actually working (1, 2, 3, or 4).

Scenarios:

1. One Nurse Working:
   → all 10 patients are served by 1 nurse
2. Two Nurses Working:
   → Partition 10 patients among 3 nurses, w/

each nurse serving at least one patient

3. Three Nurses Working:

→ partition 10 patients among 3 nurses, w/ each nurse serving at least one patient

4. Four Nurses Working:

→ partition 10 patients among 4 nurses, w/ each nurse serving at least one patient

Calculation Each Scenario: the # of ways to partition n items into k parts where each part is at least 1 size by the Stirling #'s of the Second Kind, denoted as S(n, k). This counts the # of ways to partition a set of n objects into k non-empty subsets.

→ Scenario 1: trivial, only 1 way

→ Scenario 2: S(10, 2)

→ Scenario 3: S(10, 3)

→ Scenario 4: S(10, 4)

The total # of combinations is the sum of the

Combinations in all scenarios

Calculation: We will calculate the total combinations by summing the Stirling #'s for each scenario. Let's perform this calculation

```
# Calculationg combinations for each scenario
# scenario 1 : One Nurse Working
*1 combo_1_nurses = 1 # trivial case, only 1 way
# scenario 2 : two nurses working
*2 combo_2_nurses = stirling (10, 2)
# scenario 3 : three nurses working
*3 combo_3_nurses = stirling (10, 3)
# scenario 4 : four nurses working
*4 combo_4_nurses = stirling (10, 4)
# total combos
total_combos = (*1 + *2 + *3 + *4)
total_combos.evalf() # evaluate to get a
                       numerical value
```

Alternate Approach: we can use a general combinatorial formula for distributing n indistinguimable objects (patients) into k distinguishable groups (nurses), where each group must recieve at least one object. This is often approached w/ combinations & permutations

1. Basic formula

→ the # of ways to distribute n items into k groups w/ each group recieving at least one item is $\binom{k-1}{n-1}$

2. Calculating for Each Scenario:

→ We apply this formula to each scenario (1 to 4 nurses working)

3. Total Combinations

→ the total combinations are the sum of combinations from each scenario

The total # of diff combinations for the 10 friends to be served by up to 4 nurses, where each nurse serves at least one person = 130

This calculation uses a combinatorial formula for distributing indistinguishable objects (patients) into distinguishable groups (nurses), w/ each group recieving at least one object. It provides an estimate of the diff possible arrangements of patients among the nurses, taking into account the scenarios where diff #'s of nurses are working