ECE345 HW1

Homework Group #85
Yanni Alan Alevras | yanni.alevras@mail.utoronto.ca | 1009330706
Evan Banerjee | evan.banerjee@mail.utoronto.ca | 1009682309
Gautam | gautam.makheja@mail.utoronto.ca | 1008788974

Total Pages: 10

Thursday 19$^{\text{th}}$ September, 2024

1 Question 1 Examples of this is a second test

a.

Theorem. Give a combinatorial argument to prove that

$$\sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n$$

Proof. Proof by combinatorial Argument

imagine you have a bucket with three balls in it, where each ball is either red, green, or blue.

let's say each time you pick one of the balls out of the bucket, you make a dash on a piece of paper in whatever color the ball is, and then put the ball back. Assume that you write the dashes in order so that after repeating this process n times, you end up with a sequence of n dashes in order, where each dash is either red, green, or blue.

since each dash was one of three possible choices, there are 3^n possible sequences of colored dashes that you can draw with this process.

Each colored dash has 2 properties, its color, and its index in the sequence.
For example, if n = 5, one possible order could be RGGRB. in this case, you could represent this unique order as

(G, 3), (R, 1), (G, 2), (R, 4), (B, 5)

where, in each tuple, the letter is the color and the number is the index of that color in the sequence.

Clearly the index of each colored dash is going to be a positive integer between 1 and n

Because each unique sequence of n colored dashes can be represented by a unique, unordered list of color-index tuples, any procedure that can generate this list of tuples is mathematically identical to the process of drawing the sequences by choosing colored balls.

Now imagine you had a bucket with n numbered tokens in it. Each token has number written on it ranging from 1 to n.

One possible way to create an unordered list of color-index tuples (and as a result, a way to create the original sequence of colored dashes) would be to first decide how many red tuples you wanted ahead of time, create those tuples by randomly selecting indices from the bucket, and then randomly choose between green and blue as you made the remaining tuples

Again, assuming n = 5, you could start out by saying that you will have exactly 2 red tuples. To do this, you could randomly choose 2 tokens from the bucket without replacement.

Whatever numbers you pulled out of the bucket, you would create red tuples using those numbers as your indices. For example, if the numbers you pulled out of the bucket were 1 and 4, then you would have the tuples (R, 1) and (R, 4). This exactly defines the red dashes in the sequence presented earlier in this proof.

However, there's no reason why the tokens pulled from the bucket had to be 1 and 4. Since you're choosing 2 tokens from a set of 5 tokens without replacement, there are $\binom{5}{2}$ possible combinations 2 red tuples.

After forming these red tuples, there are now 3 indices left. Each of these indices could be paired with either the color green, or the color blue.

You could choose to keep pulling numbers out of the bucket, assigning each index token that you pull out to either green or blue based on the toss of a coin

However, another equally valid approach would be to just pour out the remaining tokens, line them up in ascending order, and then flip a coin as you look at each token one by one

Heads means the index makes a tuple with green, tails means it makes a tuple with blue.

If the remaining indices after making the red tuples were 2, 3, and 5, and if the three coin tosses had given the sequence H, H, T, then we would end up with the tuples

(G, 2), (G, 3), (B, 5)

Which represents the blue and green dashes in the example shown earlier in this proof.

Because there are 2 possible outcomes for each of the 3 remaining tokens (Heads or Tails), the number of possible combinations of green and blue tuples is 2^3.

Since there are $\binom{5}{2}$ combinations of red tuples, and 2^3 combinations of green and blue tuples, there are $\binom{5}{2} * 2^3$ possible combinations of 5 index-color tuples where 2 have the color red

This means that there are $\binom{5}{2} * 2^3$ possible sequences of 5 colored dashes (from the colors red, green, and blue) where two of those dashes are red

However, if we go back to the tuple generation, there was no reason why we had to have exactly two red dashes. We could have had no red dashes, or five red dashes, or any number in between

Let's define k as the number of red tuples (where $k \in [0, 5]$). for each k, there are $\binom{5}{k}$ possible combinations of k red tuples, and since there are 5-k remaining indices, we'll end up with 2^{5-k} unique combinations or green and blue tuples. In other words, there are $\binom{5}{k} * 2^{5-k}$

Every unique combination of tuples will have some amount of red tuples, with the exact amount being between 0 and 5 inclusive.

As a result, if we were to sum up all the possible combinations of tuples as the number of red tuples (k) goes from 0 to 5, then we would end up with every possible combination of 5 tuples formed from the colors red, green, and blue.

This sum would look like:

$(\binom{5}{0} * 2^5) + (\binom{5}{1} * 2^4) + (\binom{5}{2} * 2^3) + (\binom{5}{3} * 2^2) + (\binom{5}{4} * 2^1) + (\binom{5}{5} * 2^0)$

Which is the same as:

$$\sum_{k=0}^{5} \binom{5}{k} 2^{5-k}$$

Due to the equivalency of unique tuple combinations and unique RGB color sequences, this sum is also equal to the number of unique sequences of 5 colored dashes.

However, there's no reason why we had to choose 5 as our number of tuples. This process works just as well for making n unique color-index tuples using red, green, and blue.

If we choose to make n unique color-index tuples, and we choose to make k red tuples, then there are $\binom{n}{k}$ unique combinations of red tuples, and 2^{n-k} unique combinations of green and blue tuples.

this means there are $\binom{n}{k} * 2^{5-k}$ unique combinations of n tuples where k tuples are red, and

$$\sum_{k=0}^{n} \binom{n}{k} 2^{n-k}$$

unique combinations of n color-index tuples as a whole

Again, due to the equivalency between tuple combinations and RGB sequences, this formula also describes the number of unique sequences formed from red, green, and blue dashes.

And with a little bit of algebra, it's easy to see that:

$$\begin{aligned}\binom{n}{k} &= \frac{n!}{k!(n-k)!} \\ &= \frac{n!}{(n-k)!(k)!} \\ &= \frac{n!}{(n-k)!(n-n+k)!} \\ &= \frac{n!}{(n-k)!(n-(n-k))!} \\ &= \binom{n}{n-k}\end{aligned}$$

which makes intuitive sense because the number of ways to choose k elements to use out of a set of n (without replacement) is the same as the number of ways to choose n - k elements not to use out of a set of n (without replacement)

□

b. Time complexity: $\mathcal{O}(n)$

Proof. ($\Leftarrow$) Assume by Contradiction,

($\Rightarrow$) □

2 Question 2 Example Pseudocode

```
1: function GALE-SHAPLEY(E, S)
2:     initialize all employers in E and students in S to unmatched
3:     while an unmatched employer with at least one student on its preference list remains do
4:         choose such an employer e ∈ E
5:         make offer to next student s ∈ S on e's preference list
6:         if s is unmatched then
7:             Match e with s                                  ▷ s accepts e's offer
8:         else if s prefers e to their current employer e′ then
9:             Unmatch s and e′                                ▷ s rejects e′
10:            Match e with s                                  ▷ s accepts e's offer
11:        end if
12:        cross s off e's preference list
13:    end while
14:    report the set of matched pairs as the final matching
15: end function
```

3 Question 3

Figure 1: Example image

This figure shows xxx

4 Question 4 Induction

a.

Theorem. *Some theorem here.*

Proof. Proof by induction

Base Step: When $n = 1$, is true.

Induction Hypothesis: Suppose is true

Induction Step: Consider when □

b. Time complexity: $\mathcal{O}(n)$

Proof. ($\Leftarrow$) Assume by Contradiction,

($\Rightarrow$) □

5 Question 5 Probability

a.

Theorem. *Some theorem here.*

Proof. Proof by induction

Base Step: When $n = 1$, is true.

Induction Hypothesis: Suppose is true

Induction Step: Consider when □

b. Time complexity: $\mathcal{O}(n)$

Proof. ($\Leftarrow$) Assume by Contradiction,

($\Rightarrow$) □

6 Question 6 Graphs, Proof by Contradiction

Proof. Assume by Contradiction,

Assume $d(u, w) + d(w, v) < d(u, v)$.

This implies that there is a path from u to v via w whose total distance is less than the direct distance between u and v.

However, by the definition of $d(u, v)$, it is the shortest distance between u and v. Therefore, no other path, including one through w, can have a smaller distance.

This contradiction arises from our assumption, so the assumption must be false. Hence, we conclude that $d(u, w) + d(w, v) \geq d(u, v)$. □

7 Question 7 Trees, Proof by induction

Definition: A *stable parent* is defined as a node with two leaves. Let:

- n be the number of nodes,
- l be the number of leaves,
- p be the number of stable parents.

Proof. Proof by induction.

Assume the inductive hypothesis:

$$l_n = p_n + 1 \quad \Rightarrow \quad l_{n+1} = p_{n+1} + 1$$

We will now consider two cases:

Case 1: Add a child to a leaf.

- Since the leaf gains a child, while a new leaf is added, we have $l_{n+1} = l_n$.
- Since no node with 1 child gains another child, $p_{n+1} = p_n$.
- Therefore, $l_{n+1} = l_n = p_n + 1 = p_{n+1} + 1$.

Case 2: Add a child to a node with 1 child already.

- Since no leaf stops being a leaf, you have simply added a leaf, so $l_{n+1} = l_n + 1$.
- Since the node with 1 child gains another child, a new stable parent is formed, so $p_{n+1} = p_n + 1$.
- Therefore, $l_{n+1} = l_n + 1 = p_{n+1} + 1 = p_n + 1 + 1$.

Base case: For $n = 1$, we have:

- 1 node implies 1 leaf and 0 stable parents.
- Thus, $l_1 = 1$ and $p_1 = 0$.
- Therefore, $1 = 0 + 1$, which satisfies $l_1 = p_1 + 1$.

Q.E.D.