CS 315 - Day 07, Series Algebra

We've got four students: Jim, Jill, Jon, and Jer.

How many ways can we pick two of them? It doesn't matter what order they are chosen, and there are no repeats.

Jim, Jill
Jim, Jon
Jim, Jer
Jill, Jon
Jill, Jer
Jon, Jer

We have an easier way to calculate this, using a combination. $C(4,2) = \binom{4}{2} = \frac{4!}{(4-2)!\cdot 2!} = \frac{4\cdot 3\cdot 2\cdot 1}{(2\cdot 1)\cdot(2\cdot 1)} = \frac{4\cdot 3}{2\cdot 1} = 6$

In general, the formula for combinations is $C(n,k) = \binom{n}{k} = \frac{n!}{(n-k)!\cdot k!}$

Last class we proved a quantified predicate involving handshakes,

$P(n)$: *n students in class will result in* $\frac{n(n-1)}{2}$ *handshakes, for at least* $n = 2$ *students.*

We can frame this problem differently, using combinations. In a group of $n$ people, how many ways can 2 shake hands?

$$C(n,2) = \binom{n}{2} = \frac{n!}{(n-2)!\cdot 2!} = \frac{n\cdot(n-1)\cdot(n-2)!}{(n-2)!\cdot 2!} = \frac{n\cdot(n-1)}{2}$$

Which is the same as the right-hand side of the predicate we proved previously.

Note that the following are ***closed*** formulas, the since RHS can be calculated directly:

$$\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$$

$$\sum_{k=1}^{N} k^2 = \frac{N(N+1)(2N+1)}{6}$$

Let's review some associated series algebra. What if you used the top sum and embedded it so $k$ was incremented?

$$\sum_{k=1}^{N}\sum_{i=1}^{k} i = \sum_{k=1}^{N}\frac{k(k+1)}{2} = \sum_{k=1}^{N}\frac{k^2+k}{2} = \frac{1}{2}\sum_{k=1}^{N}(k^2+k) = \frac{1}{2}\sum_{k=1}^{N}k^2 + \frac{1}{2}\sum_{k=1}^{N}k$$

Substitute the RHS of the first equation  Distribute  Factor out the $\frac{1}{2}$  since adding, split into two sums

An application of this is in the "Twelve Days of Christmas" example,
https://www.pnc.com/en/about-pnc/topics/pnc-christmas-price-index.html

In-class w/***assigned groups of four (4)***, write 0. and 1. on board as ground rule(s):

0. No devices.
1. Intro: Name, POE, Icebreaker.
2. Pick someone on the group who will report out.
3. Work the following:

Remember, the sigma is just a nickname.

For the following, write the terms out in expanded notation:

$$\sum_{k=1}^{5} 3k^2 + k$$

Then try grouping the terms that are the same, the squares and so on:

$$\sum_{k=1}^{5} 3k^2 + k = (3\cdot 1^2 + 1) + (3\cdot 2^2 + 2) + (3\cdot 3^2 + 3) + (3\cdot 4^2 + 4) + (3\cdot 5^2 + 5)$$

$$= 3\cdot 1^2 + 3\cdot 2^2 + +3\cdot 3^2 + 3\cdot 4^2 + 3\cdot 5^2 + 1 + 2 + 3 + 4 + 5$$

$$= 3\cdot(1^2 + 2^2 + +3^2 + 4^2 + 5^2) + 1 + 2 + 3 + 4 + 5$$

Let's convert these back into sigma notation:

$$3\sum_{k=1}^{5} k^2 + \sum_{k=1}^{5} k$$

So, the point is that if you're adding terms in a summation, you can easily split the summation. And you're able to factor constants out of the sigma/summation.

Now, only one minor change, the constant $5$ becomes an $n$

$$\sum_{k=1}^{n} 3k^2 + k$$

Perform a similar simplification.

$$\sum_{k=1}^{n} 3k^2 + k = 3\sum_{k=1}^{n} k^2 + \sum_{k=1}^{n} k$$

So, for this class, what expectations do we have regarding Induction and Series?

- You are comfortable reading and performing Inductive proofs.
- You are comfortable manipulating series algebraically, with or without "sigma" notation.

The textbook uses Random Access Machine (RAM) model, not to be confused with RAM memory.

In this model, each instruction and data access takes a ***constant*** amount of time:

- Instructions execute sequentially, one after another, no concurrency.
- Each instruction takes the same time to execute (there isn't a 'miracle' command).
- Each data access takes the same amount of time.

$O()$ notation is what was covered in CS 240 – Computer Science 2, and it's the "industry standard."

But mathematically, and sometimes it's used imprecisely. In CS 240, we said, "the best possible O()," which is basically Θ() , or "Omega."

- $\Theta()$ – tight bound
- $\Omega()$ – lower bound
- $O()$ – upper bound

Pseudo-code conventions:

- Indentation indicates block structure.
- Looping conventions similar to C/C++, Java, Python, and Javascript.
- Comments are preceded by # or //.
- Variables are local to the given procedure/function.
- Access array elements with brackets, such as, A[1].
- ***Following the convention of the textbook***, the arrays will start at 1, NOT 0.