Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript English

0:03
All right, now you have an implementation of the rather oversimplified version of this program. It just looks for ATG and then TAA, and gives back everything in between.

0:14
But real genes have to be a multiple of 3 in length, since they're made of codons. Each of which is 3 nucleotides long.

0:24
For example, this string is a valid gene.

0:28
You can see here how it can be divided into codons starting with ATG and ending with TAA.

0:35
However this string is not valid.

0:38
Even though it has ATG and it has TAA when we look between them we don't find a valid sequence of codons.

0:53
Now, let's make our algorithm a little bit more realistic. You will fix this aspect of your algorithm, it will still be a simplification, but a bit more realistic. At this point, it's really good to note that starting with a simple version and adding features, is not only a useful technique, for you, as you learn new concepts. It let's us introduce a few concepts at a time. But also important when writing real, large complex problems.

1:24
Okay, so you just saw two examples of DNA strings.

1:29
One that has a valid gene and one that does not. How can you tell, in general? What is the change you need to make to your algorithm?

1:42
It might be useful to show the indices as we have shown here and highlight the location of the start and stop codons. Do you see how to algorhythmically tell the difference? If not, that's fine. It can be hard to spot patterns. But you will get better at it with practice.

2:04
One great technique to find patterns is to make a table. You might remember that we did this in some of our examples. We can then add more examples to our table as I'm showing here to help us see the pattern. Some with yes answers and some with no answers.

2:23
Maybe you see the pattern now or maybe it is still hard to see.

2:27
What can you do for pattern still isn't clear? Maybe adding more rows will help or maybe not.

2:35
Instead we might start exploring the relationships between the items in the table. Here we have added another column for the difference in the index of the stop codon and the index of the start codon.

2:48
The ones that have yes answers have differences of 6 and 12. And the ones that have no answers have differences of 4 and 11. What do 12 and 6 have in common that 11 and 4 do not? We might be able to think of a lot of things, 6 and 12 are both multiples of 6. But that doesn't make sense in the context of this problem. If we did more examples, we could find ones with yes answers that have 3 or 9 or 15.

3:18
However, 6 and 12 as well as 3, 9 and 15, are all multiples of 3. This relationship does make sense. As we know that the length must be a multiple of 3. Now that you know the relationship, you know that you want to look for. You need to do some math in Java. You need to a way to ask if the difference between two numbers is a multiple of three.

3:42
If you took course one with us you might remember the mod operator which gives you the remainder when you do division. x mod y written x % y means divide by, divide x by y, but give me the remainder, not the quotient. This can help you with the problem at hand since a number that is a multiple of three has a remainder of zero when divided by three. That is, if x minus three is equal to zero, then x is a multiple of three. You can use other mathematical operators in Java, + minus, times, divide, they're all valid.

4:18
You can also use equals equals to see if two numbers are the same. Not equals to see if they're different and less than, less than, or equals, greater than, or greater than or equals to check for inequalities. You can also combine simpler expressions into more complex expressions. This expression checks if (a- b) mod 3 is equal to 0. It is evaluated by first evaluating (a- b), then taking that result and doing mod 3 on it. Then finally taking that result and checking if it's equal to 0. This is pretty much exactly what you need for the prominent hand. To see if the difference between two things is a multiple of three.

5:04
While you are learning about math in Java, there are several different types of numbers, actually there are some variations on these. But, you don't need to worry about that right now.

5:17
One type of number is int, which represents integer numbers like -2, -1, 0, 1, 2, etc. Ints can't have a fractional part. The other type is double, which represents real numbers. Ones with fractional parts like 1.2 or 3.4, 5.7, of course you could also represent 3 which is 3.0 with a double. You only want to use doubles when you need to since they have some behaviors that can be really confusing to novice programmers.

5:54
One word of caution about integers however is that math on integers always yields integers. So what do you get if you divide 5 by 2? If you are thinking 2.5, remember that you can only get an integer result. So you get 2.

6:12
What about 100 divided by 3 times 4? You will get 132 since 100 divided by 3 is 33 and 33 times 4 is 132. So what if you do 100 times 4 divided by 3? Seems like you should get the same answer right? Well actually now you get 133, why? 100 times 4 is 400 and 400 divided by 3 is a 133. These sorts of issues should not come up in these courses, but are good to be wary of if you're doing integer division.

6:52
Another thing to know about math in Java is that it has order of operations. Rules like math in programmer speak these rules are called precedents and associativity.

7:04
As with math, a plus b times c means to do b times c first then add that result to a.

7:12
What if you have mod? It has the same precedence as division, meaning it takes place at the same place in the order of operations. So a- b mod 3 means to (b mod 3), then subtract that result from a. This is why we put parentheses around a- b earlier when we wanted to do the minus first, then take its result mod 3.

7:35
Comparisons for equality happened very late in the order of operations. A plus B equals C minus D means to first do A plus B then do C minus D and finally compare the two results to see if they are equal to each other.

7:50
These rules are typically like they are in math. Multiplication and division comes first. Then addition and subtraction. Also like math you can use parentheses to group things so that they come first. If you're not sure what order things will happen you can always use parentheses to be explicit and be sure you get what you want. Okay, now that you know about math in Java, it's time to go improve your gene finding algorithm. Thank you.

Downloads

Lecture Video mp4
Subtitles (English) WebVTT
Transcript (English) txt

Would you like to help us translate the transcript and subtitles into additional languages?