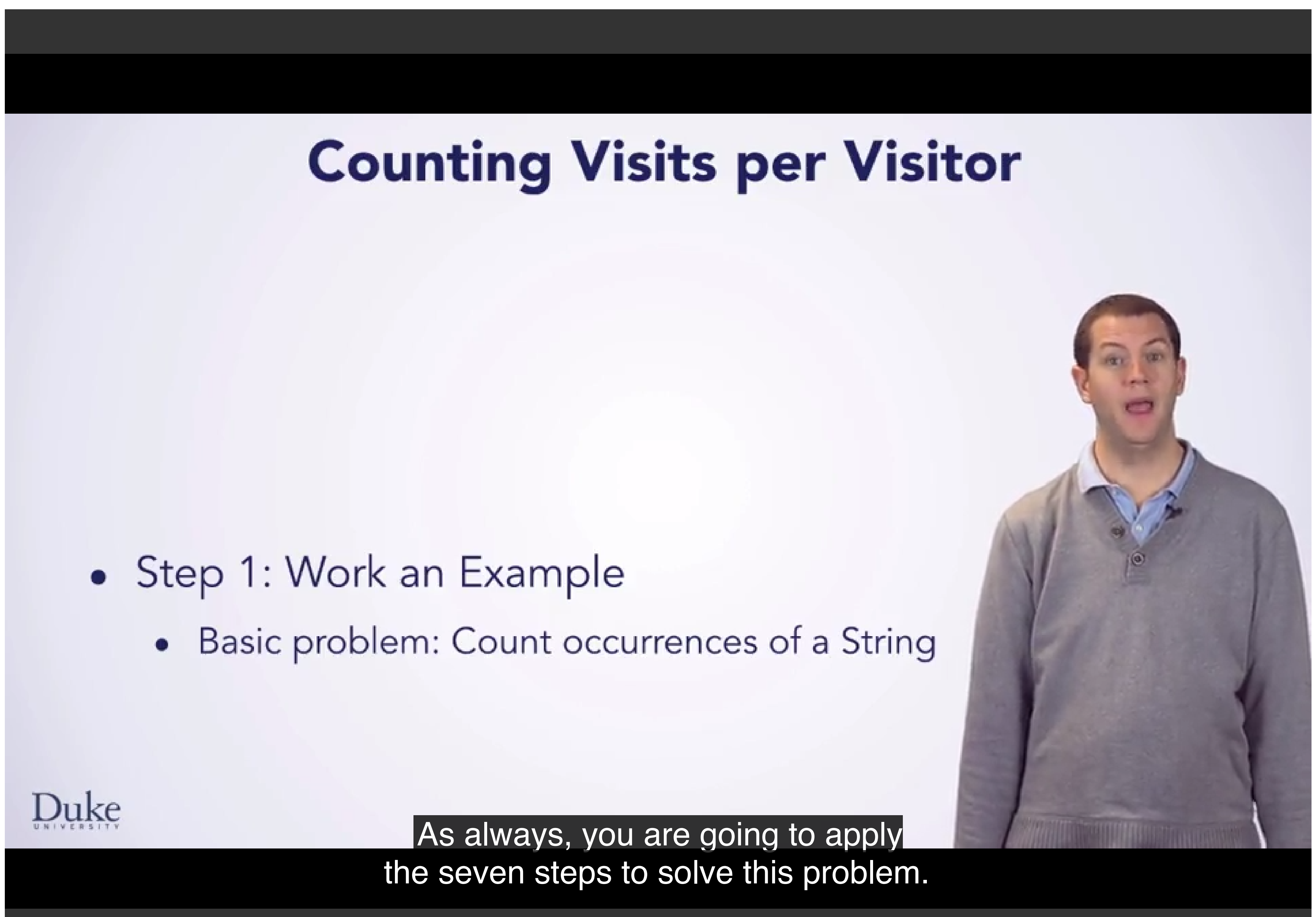
Counting Visits per Visitor
Step 1: Work an Example
Basic problem: Count occurrences of a String
Duke
As always, you are going to apply the seven steps to solve this problem.

Developing an Algorithm

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

Interactive Transcript

Search Transcript English

0:03
As always, you are going to apply the seven steps to solve this problem. As with the previous problem, we'll observe that the problem is fundamentally the same no matter what Strings you use. And use Strings that are easier to talk about than IP addresses. We'll work through this using some animals. We want to count how many times each animal name appears in this list, cat, snake, T-Rex, snake, cat. Now, you start by working through this example in a step by step fashion, examining each animal and keeping count of how many times you have seen its name. When you have finished looking at each animal in the list, you have your answer. Cat and snake both appear twice, and T-Rex appears once. Now that you've worked an example, it is time to think about and write down exactly what you just did. The first thing was to make an empty table, or you can keep track of each name and how many times you had seen it. Saying you made a table is fine, but it is also good to think about what this means in terms of actual data types you can use in your program. What type have you seen that is useful for representing this kind of information?

1:14
Yes, a HashMap that maps Strings names to Integers counts.

1:20
Once you realized that this is a HashMap, you may as well call the columns by their technical names, Key and Value.

1:28
Finally, you'd want to give this HashMap a name so that you can refer to it easily. We'll call it Counts.

1:35
Next, you looked at the first string in the list, Cat. Then you looked for Cat in counts and saw it wasn't there, so you put it in with a value of one. You did a similar step for the second String Snake, which was also not in counts, and for the third String, T-Rex. For the fourth string, Snake, things were a little different. When you look in count, you see Snake is already there with a count of one, so you update the count to be two. And similarly, for the last string, Cat, you find that it already has a count of one. So, you update it to have a count of two.

2:11
After all of that, the entire HashMap count is your answer.

2:17
That leads to these steps for this particular instance of the problem. So now, it is time to find patterns and generalize to any instance of the problem. Notice there are several steps where you did not find the current String in the HashMap. In each of these cases, you put it into the HashMap with a value of one. Why one? Do you always want one, or should you look for some other pattern? If you think about it for a moment, you will realize that here, you always want one. It is the first occurrence of that name, so you've seen it once.

2:49
There are also some cases where you already had that particular name in a HashMap. In these cases, you updated the value to be 2. Again, you should should ask yourself, why did I use 2? Do I always want 2, or is there some other pattern? In this case, you do not always want 2. Instead, you want the old value plus 1. That just happens to always be 2 here, because of the specific example you worked. With all of that in mind, you should be able to generalize these steps and come up with an algorithm that looks like this. This algorithm makes an empty HashMap, then iterates over each String in the input, and checks if that String is already in the HashMap. If not, the algorithm puts that String in to the HashMap with a value of 1, and if it's already there, it updates the value to be 1 more than the old value. After processing all the Strings, the answer is the HashMap counts.

3:44
Now, you want to test this algorithm out. Test it on the input fish, dog, fish, fish.

3:53
This algorithm got you the right answer so you can be more confident that it's correct.

3:58
Before you turn this into code, we're going to remember that our input is not actually a list of Strings but a list of login entries whose IP addresses you want to process. This slightly adjusted algorithm is basically the same. But we have changed this variable name to le, to stand for log entry, and are iterating over the contents of the array list records, which is an instance variable in the LogAnalyzer class. Then, you want to get the IP address out of that log entry, and use that as the string that you use to update the HashMap. Now, it is time to turn this into code.

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

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