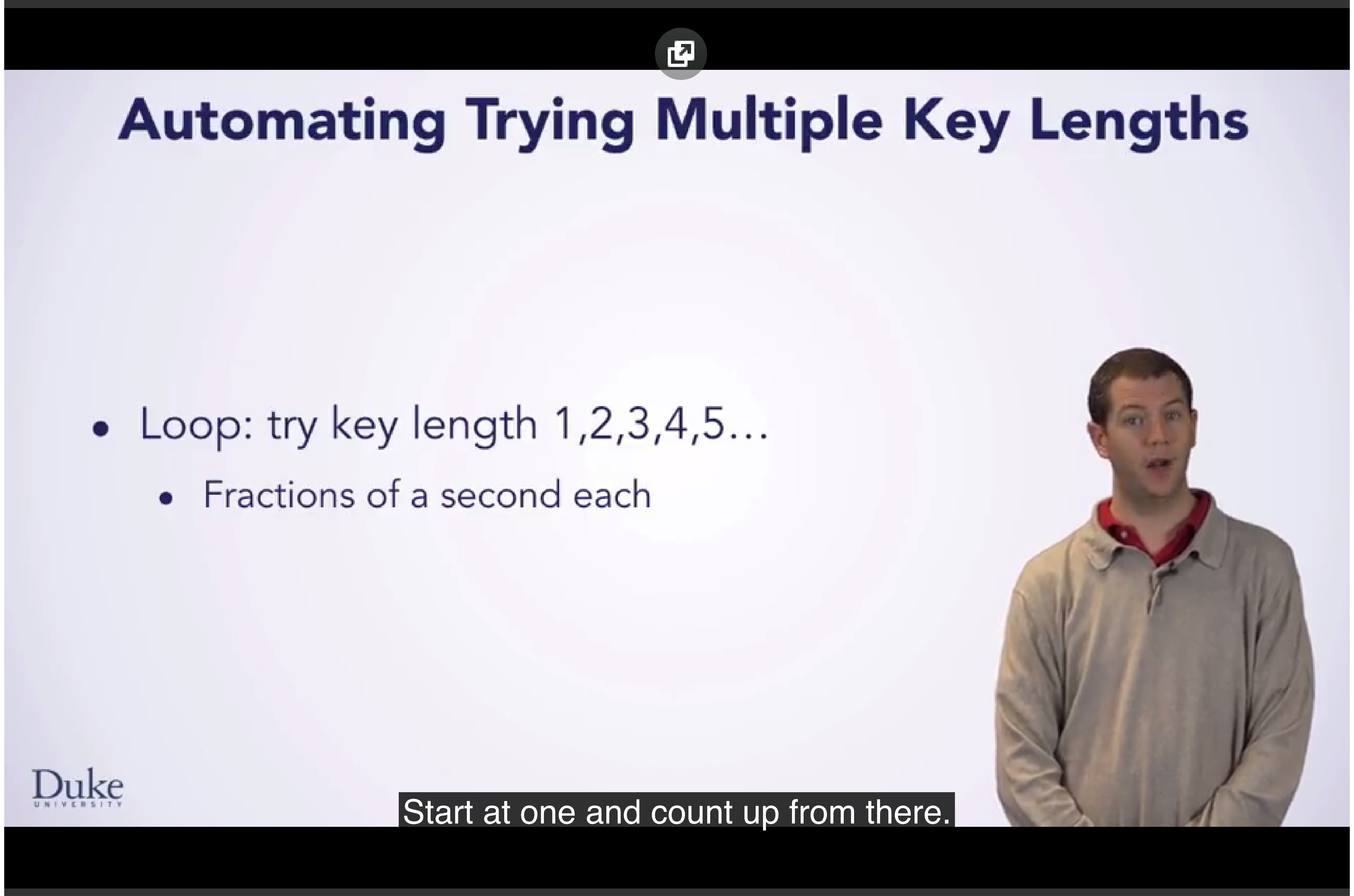
Breaking the Vigenère Cipher


Automating Trying Multiple Key Lengths
• Loop: try key length 1,2,3,4,5…
• Fractions of a second each
Duke
Start at one and count up from there.

Unknown Key Length

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

Interactive Transcript

Search Transcript English

0:03
Now you've written a code to break the vigenere if you know the key length. But what do you do if you don't know the key length? Well, you could just try some different key lengths. You could use the code you just wrote and pass it a key length, then see what the result is. Maybe the key length is one.

0:21
That output is incomprehensible. One is probably not the correct key length. What if you tried two?

0:30
That doesn't seem to be right either. How about three?

0:36
Oh, hey. This output looks like it could be English. It has readable words. It would seem this message was encrypted with a key length of three. You could write a loop to try many different keylengths. Start at one and count up from there. The computer can try one particular key length in a fraction of a second. So, even if a message which is thousands of lines long is encrypted with a keylength of 92, the program can do it in no time but how do you tell if the key length is right? Do you really want to look at the output for each iteration to say? That's what we did a moment ago. Looked at the output to see if it was meaningful text.

1:14
But maybe if we think carefully about what we just did, we can find a way to automate it.

1:20
As you look at the incorrect decryption here, think about how you knew it was not right. This group of three letters forms a word but J-O-W is not a real word. Likewise y-t isn't a real word nor is Y-O-B. None of the words in this message are actual English words.

1:43
Contrast that with the correct decryption. H-O-W is the word how. D-O is do and Y-O-U is you. All of the words in the correct decryption are actual words. This observation leads to the idea of how to figure out which key length is right. Count the number of real words in the output. You would start by reading a list of English words from a file. You could store the list of words in an array list but we're going to recommend you use a hash set, which will give you the same functionality but work much faster.

2:20
Then, you try the decryption for various key lengths. Start at one and count up. Where do you end? Well you could just count up to the message length. Although if the key is close to the length of the message, you won't be able to break it because there won't be enough letters to meaningfully frequency count. For this mini project, just count up to 100 then for each key length you try, see how many of the words in the decrypted text are actual words from the file you read in.

2:52
Once you have done that, choose the key length, key, and decrypted text which give you the most real words. As we just mentioned, using an ArrayList would work just fine. As you read in the file, you would use the .add() method to put each word in the list. When you want to see if a potential word is actually a real word from the list, you would use the .contains() method. A better option is to use the HashSet class. Like ArrayList and HashMap you can use a HashSet containing different types of data. So, you need to put String in angle brackets after HashSet. You want a HashSet of strings. For this problem you will use the HashSet in much the same way as you would an ArrayList. You can call .add and .contains on it. You cannot index into a HashSet like you can an ArrayList but if you wanted to iterate over one, you could do that with a for each loop. That is what has been happening behind the scenes when you iterate over a hash maps.key set. The main advantage is that dot contains will be much faster. Instead of searching through every word in the hash set, it can look at only a few words to figure out if it contains the requested information or not. The last thing that we will mention before you start coding is how to split a string up into individual words. You've already seen that String has a .split() method. You can use that passing in quote \\W quote. This asks the split method to divide the string up at every character that is not part of a word meaning spaces, punctuation or numbers. Once you have done that you can iterate over those words with a for each loop, like this one.

Downloads

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

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