

Lesson 9: Lossless Compression

Overview

Students use the Text Compression Widget to experiment with compressing songs and poems and try to find their 'personal best' compression. A video introduces important vocabulary for the lesson and demonstrates the full features of the widget. Students pick a text they think will be 'easy' to compress and one they think will be 'difficult', paying attention to why some texts might be more compressible than others. As a wrap-up, students discuss what factors make some texts more compressible than others.

Purpose

As students have been creating images over the last few lessons, the number of bits it takes to represent that information has grown and grown. In this lesson, students are introduced to the concept of compression as a way to address the growing file sizes of all of our information. This lesson is anchored by the Text Compression widget, which is a very hands-on & active widget for students to experiment with. Most of the lesson should be spent in the widget, having students experiment with different strategies for compression and creating a memorable experience to help anchor the concept of compression. Students also watch a video that introduces **lossless** and **lossy compression** - today's lesson is an example of **lossless compression**, while tomorrow's lesson is dedicated to **lossy compression**. The widget is just one example of lossless compression and students aren't expected to master specific compression strategies - instead, they should understand that lossless compression uses less data and still lets them re-create the original information.

Standards

Full Course Alignment

CSP Conceptual Framework

- **DAT-1** - The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily understood by people.

CSTA K-12 Computer Science Standards (2017)

- **DA** - Data & Analysis

Objectives

Students will be able to:

- Create lossless compressions of text files
- Analyze patterns in data to determine compression strategies

Preparation

- Familiarize yourself with the Text Compression Widget
- Open Unit 1 Slideshow to current lesson
- Plan for how you will display the initial Pitter Patter Message & it's Compressed Message during the warm-up

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the teachers

- **CSP Unit 1 - Digital Information** - Slides

For the students

- **Text Compression widget (tutorial)** - Video (**Download**)

Agenda

Lesson Modifications

Warm Up (5 minutes)

Activity (35 minutes)

Introduction to Compression (5 Minutes)

Text Compression Widget (15 Minutes)

Comparing Compressions (10 Minutes)

Wrap Up (5 minutes)

Synthesis

Assessment: Check For Understanding

Teaching Guide

Lesson Modifications



Attention, teachers! If you are teaching virtually or in a socially-distanced classroom, please read the full lesson plan below, then click **here** to access the modifications.

Warm Up (5 minutes)

 Have students list additional abbreviations they might use in their daily lives.


 **Discuss:** *Why might we use abbreviations when sending messages? What are the advantages?*

Discussion Goal: There are many possible responses to this - to talk in code, to hide information, to be clever - but an important response to highlight is that abbreviations save time & space when communicating. If a student suggested an abbreviation that not everyone knew, this is a great moment to bring up that both the sender and the receiver need to understand what the abbreviation stands for in order for it to make sense. Both of these points foreshadow today's activity on compression.


Activity (35 minutes)

Introduction to Compression (5 Minutes)

Remarks

 I want to send this message to a friend, but their phone can only accept 80 characters of text at a time. I notice this pattern has some repetition in it, so rather than sending the whole message, I send this instead:

 **Do This:** Click to show next slide with message highlighted in red.

 **Discuss:** *How is this message the same as the first? What actually gets sent to my friend?*


Discussion Goal: Students should notice that each symbol represents other snippets of text. By substituting each symbol for the text it represents, we can re-create the original message.

Students may need some guidance to see that the entire sent message is really two parts - the text with symbols and the key that shows what each symbol represents. Students should see that both need to be sent in order for the original message to be recreated - if only the text is sent, the receiver won't know how what each symbol represents to recreate the message.

Remarks

Using abbreviations and symbols is a form of **compression**, where we try to represent the same information with fewer characters. The original message had 93 characters, but the new message and key, also called a dictionary, have a total of 56 characters. We're essentially sending the same information, but with fewer characters. Our goal today will be to create our own text compressions using similar methods.

Text Compression Widget (15 Minutes)

 **Do This:** Have students log into Code Studio and open Level 1 of this lesson - the Text Compression Widget.


 1

Widget: Text Compression

Remarks

This widget will let you use symbols to compress the text in the center of the screen. You can type in the dictionary on the right-side. As you do, the text on the left-side will update with your symbols. You have 4 minutes to try and compress this text as best you can.

Circulate: Help students understand how this widget works so they can successfully compress text. Make note of students who have found successful strategies so they can be highlighted in the upcoming discussion.


 **Regroup:** Gather the class back together. Emphasize the black box at the bottom of the widget, which has their current compression rating. Have students make a note of their current Compression Percentage at the bottom of the box

 **Discuss:** *What strategies are you using to compress your sample text? Which ones seem most successful?*

Discussion Goal: Students will have encountered a variety of strategies, but there are a few worth emphasizing for the full class:

- Look for repeated words, sentences, or even parts of words (like -ing or -th).
- The widget lets you copy-and-paste to embed symbols within symbols. This was demonstrated in the pitter-patter example where some symbols were 'unpacked' to include other symbols.
- The order of the dictionary matters and trying to rearrange the dictionary once it's made can lead to problems

It's okay if not all of these points are highlighted - the upcoming video will emphasize these same points for students.

 **Video:** Show (feel free to skip from 2:30-5:00 if your students are comfortable with how the widget works, but don't miss 5:00+). After the video, be sure to emphasize two things:

- The widget we are using is an example of **lossless compression**

- The compression percentage at the bottom of the screen is calculated by comparing the number of **bytes** in the original message and the number of **bytes** in the compressed message.

Text Compression widget (tutorial)



Video: Text Compression Widget with Aloe Blacc

Do This: Give students another 4 minutes to apply the strategies they've just seen to continue to raise their compression percentage.

Circulate: Check in with students on their strategies and their compression rates. Encourage students to continually try and reach a 'personal best' by looking at how their compression rates change when they add or remove items from the dictionary.

Remarks

We're starting to reach the 'limit' for how much we can compress this particular message. But not every message can be compressed with a high rating. We're going to investigate what makes some messages more compressible than others.

Teaching Tip

Competitions: You could incorporate a peer-to-peer competition (in small groups or as a full class) to get the 'highest' rating, but that can be isolating for students and suggests there is a single 'best' way to do this. An alternate strategy is: when students start for the second time, have them compete against themselves to beat their rating during the first 4 minutes. In this way, success is measured by personal growth and has a higher chance of letting every student feel successful.

Starting Over: When solving computational problems, it can sometimes be helpful to restart completely from the beginning. This activity may be a good place to suggest this to students, especially those that feel particularly stuck or frustrated - sometimes restarting from the very beginning surfaces new ideas and strategies that we didn't see before.

Comparing Compressions (10 Minutes)

Remarks

Click the Drop-Down Menu to explore other texts to compress. Be looking for texts you predict will be 'easy' to compress and texts you predict will be 'difficult'

Group: Have students work with their neighbor for this activity. Place students in groups of 2 with at most one group of 3.

Do This: Students work together to compress an 'easy' text and a 'difficult' text.

Circulate: Give students several minutes to complete their compressions. Emphasize that students should work in pairs using two different computers so both examples can be referenced during the wrap-up discussion.

Teaching Tip


"aaaa...aaa": Many groups will probably attempt the last option, all A's, as their 'easy' text - it's possible to get a compression rating into the mid-80's with this text. This is fine, since it still emphasizes one of the big takeaways from this activity: information with high repetition is easier to

compress. However, it is also reasonable to ask groups to do a second 'easy' text once they're satisfied with this one

Priorities: It's not necessary for all groups to pick the same texts, nor is it important to find the very 'best' compressions. Instead, students should focus on the qualities that they think make some texts 'easier' or more 'difficult' than others. You can emphasize this with the questions you ask as you circulate to groups: "What made you pick this for your 'easy' text? What made you pick this for your 'difficult' text?"

Wrap Up (5 minutes)

Synthesis

 **Discuss:** *What made some messages "easier" to compress than others? What made some messages more "difficult" to compress than others?*

Discussion Goal:

- 'Easier' texts usually had lots of repetition - repeated words or phrases or syllables. A useful strategy is to use this repetition to create the compression.
- 'Difficult' texts usually have less repetition, making it less likely to apply this particular method of compression. Some strategies may actually make compression worse, which can be counter-intuitive

Remarks

There are many strategies we can use when creating **lossless compressions** and there isn't a single best way to do it. Instead, our compression rate usually depends on which strategy we choose and the patterns in the text we're compressing. Most importantly, even though the number of **bytes** is getting smaller, we're never actually losing information - we can always perfectly recreate the original message using our dictionary key.

 **Journal:** Have students add the definition of **lossless compression** to their journal

Assessment: Check For Understanding

Check For Understanding Question(s) and solutions can be found in each lesson on Code Studio. These questions can be used for an exit ticket.

Question: What is the most important quality of lossless compression?

Question: An author is preparing to send their book to a publisher as an email attachment. The file on their computer is 1000 bytes. When they attach the file to their email, it shows as 750 bytes. The author gets very upset because they are concerned that part of their book has been deleted by the email address. If you could talk to this author, how would you explain what is happening to their book?

 3-4

Check For Understanding

