

# Entropy & Information

---

## *Pre-Lab: Maxwell and His Demon*

### **A Bit of History**

We begin this lab with the story of a demon. Imagine a box filled with two types of gas molecules. We'll call them blue and green. A solid partition is inserted down the middle of the box, dividing it into two equal volumes. Then a small door that is just big enough for a molecule to pass through when it's opened is built in the partition. Finally, a "very observant and neat-fingered being"<sup>1</sup> is assigned to watch the door. His job is to sort the gas, opening the door to let the green molecules move to the right side of the container and the blue molecules to the left side of the container. (You may also read that the demon separates hot and cold gas rather than two different types of gas.) Since he is observant and neat-fingered, the being performs his duty admirably and successfully.

This being has come to be known as "Maxwell's demon", since it was James Clerk Maxwell who came up with this thought-experiment in 1867. The reason we still discuss it today is that it helped to revolutionize the way people think about information.

The Second Law of Thermodynamics states that the entropy of the universe increases with time. Right now we can think of entropy as *disorder*. Clearly, as the demon sorts the gas, the gas becomes more ordered, meaning its entropy decreases. This is okay, as long as there is an increase in entropy elsewhere. But where? The big leap was to realize that in order to perform his task, the demon must gather information. Information, it turns out, is a real physical quantity that is associated with entropy. The increase in the entropy of the universe actually comes when the demon forgets what he has learned! The proof of this fact was given by Charles Bennett just over 100 years after Maxwell first posed the problem.<sup>2</sup>

Longtime Wash U Physics Professor Edwin Jaynes was one of the pioneers in the field of information theory and its connections to statistical mechanics. You can find links to two of his seminal papers as well as a short biography on the lab website.



### **Three Interpretations of Entropy**

Entropy is often introduced (as it was in the previous paragraph) as a measure of the *disorder* of a system. This simple interpretation certainly has some appeal, but it can sometimes lead to confusion. A more rigorous definition relates entropy of a system to the *number of microstates* that can produce the macrostate of the system, also known as the *multiplicity* of the macrostate. [See Moore T3-T5 for a detailed introduction to entropy.] In this lab, we will use both of these interpretations of entropy.

Additionally, we will introduce a third interpretation of entropy. We can say that entropy represents our *ignorance of a system*. More entropy implies more ignorance. More ignorance implies more entropy. We can easily predict things about low-entropy systems. We cannot predict much about high-entropy

systems. This third interpretation certainly does not contradict the microstate interpretation: if more microstates are available, we are more ignorant about which particular microstate is actually occupied. Thinking of entropy as a measure of ignorance can be helpful in making the connection between thermodynamics and information theory, a connection that was inspired by Maxwell's demon.

## The Story

Have you ever wondered how your computer can take a large file and compress it into a byte-sized morsel? Using our three interpretations of entropy, you are going to discover some incredible connections among the worlds of thermodynamics, computation, and linguistics. We will begin conceptually, but we will conclude with an incredible face-off between you and a computer (not unlike the climax of 1983's *WarGames*).

## A Maxwell's Demon Game

As mentioned in the Bit of History, Maxwell's demon led to a revolution in the way scientists approach the concept of information. It is one of the greatest thought-experiments of all time, and it is now one of the [not-so-]greatest computer games!

**Do This:** So that you never forget what Maxwell's demon is, take a little time to play the game. You can find one version of the game on the Entropy page of the lab website under the Pre-Lab links. If the game doesn't appear, re-sizing the browser window sometimes helps. The goal is to sort the gas as described in the introduction. Click the screen to open the red door. Try to get the green gas molecules to the right and the blue molecules to the left. You can set the number of pairs of molecules to any value between 1 and 10. If you can't get this particular game to work, there are lots of similar games available on the web.

**Read This:** Answer the following questions about Maxwell's demon.

PL1. Using the interpretation of entropy as disorder, state the demon's effect on the entropy of the gas.

PL2. Using the definition of entropy as the number of available microstates, state the demon's effect on the entropy of the gas. (You could refer to Example T8.2 in Moore.)

**Read This:** To use the interpretation of entropy as ignorance, we might say something like this: when the gas is not sorted, a green molecule could be anywhere in the box. However, once sorted, we *know* that a green molecule must be on the right side. We are less ignorant about the location of any gas molecule after the gas is sorted. Since we are less ignorant about the sorted gas, its entropy has been reduced.

PL3. By invoking the Second Law of Thermodynamics, argue that entropy must increase somewhere outside of the gas.

**Read This:** Though the demon in Maxwell's thought-experiment seems pretty non-physical, it turns out that there are natural demons lurking throughout your body. Certain enzymes in your cell membranes sort ions much like Maxwell's demon sorts gases.<sup>3</sup> To successfully sort ions, these enzymes burn ATP, a messy chemical reaction which creates enough entropy to comply with the Second Law of Thermodynamics.

### A Real Demon

The March 2011 issue of *Scientific American* contains an article detailing a recently devised Maxwell's demon ("Demons, Entropy and the Quest for Absolute Zero" by Mark G. Raizen). In addition, the article touches on numerous other topics that you have learned about in Physics 198. As such, it's the perfect article to read both as an introduction to this lab as well as a review of this semester. It turns out that Washington University has a subscription to *Scientific American* that you can use. Here's how...

- If you are logged onto a Wash U network, you can go directly to the article by using the Pre-Lab link on the lab website.
- If you are using an outside network or have trouble using the above hyperlink, follow these instructions (if you're using a Wash U server, you will automatically skip step f):
  - a) Go to <http://catalog.wustl.edu>
  - b) Do a Journal Title search for "Scientific American Online".
  - c) You should get two results. Click on the first one.
  - d) Click the link for "WUSTL full text" that is in the bluish banner roughly halfway down the page.
  - e) Click the link for "Nature Journals Online".
  - f) Enter your WUSTL key login information
  - g) Scroll down the page and click on the link for the March 2011 issue.
  - h) Scroll almost all of the way down the page and open the PDF version of "Demons, Entropy and the Quest for Absolute Zero".

**Do This:** Read the article. Pay special attention to the wide range of topics that are addressed in the article that have also been addressed in Physics 198. Then answer the following questions about the article.

PL4. Other than entropy and the Second Law of Thermodynamics, list three *specific* topics from Physics 198 that are addressed in the article. (An answer like "quantum mechanics" is not specific.)

PL5. Who was the scientist that first proposed that there is entropy associated with the information that Maxwell's demon must gather as it sorts a gas?

PL6. In the real-life Maxwell's demon, an atom can get trapped after scattering a photon. Becoming trapped lowers the entropy of the atom. By the Second Law of Thermodynamics, entropy must increase elsewhere. How does that happen?

## Part I: Compressing Images – Crystals & Glasses

You may not feel like you know anything about how to compress files on a computer, but what you know about thermodynamics is actually a good start. The big idea behind the compression of files on a computer is to use patterns within the file to store it in a smaller way.

Some compression schemes such as JPEG are called *lossy*. These compressed files cannot be used to reconstruct the original file exactly. They can only approximate the original file. Other compression schemes like ZIP are *lossless*. All of the information in the original file is preserved. In this lab we will be looking at lossless compression, specifically ZIP.

The most painless introduction to compression is probably in the context of an image. The particular images we choose will also be easy to connect to thermodynamics, entropy, ignorance, and information.

### Equipment

- Computer
- crystal.bmp
- glass.bmp

### 1. File Sizes of Images

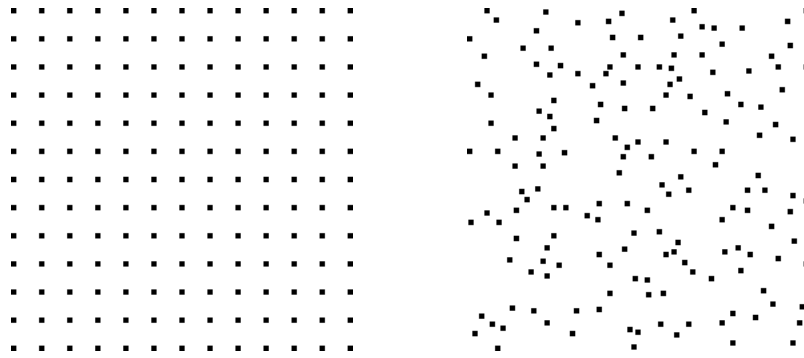
In this section, you will go through a series of steps to investigate the following claim:

**Files with more entropy are harder to compress.**

Here, you will be working with two bitmap (.bmp) files. In these bitmaps, every pixel of an image is stored using three bytes: a red byte, a green byte, and a blue byte. (Some other bitmaps will have four bytes per pixel.) So, an image with 250,000 pixels (500 x 500) takes about a 750 kB to store. But while the bitmap can be explained in a small number of words, it tends to lead to unnecessarily large file sizes. Let's learn a little about how we can make the file smaller.

**Do This:** Download the two images from the In-Lab Links page. Open both images and take a look. The two images should look a lot like those shown in Figure 1. It should be clear why each file was given its name. In both images, atoms are represented by 4-pixel-by-4-pixel squares. Each image contains 169 of these squares against a 256-pixel-by-256-pixel white background. The only difference is the arrangement of the squares. One is nicely ordered like a crystal, while the other is random like a glass. See Appendix A for a close-up view of *crystal.bmp*.

**Checkpoint 1.1:** Based on the interpretation of entropy as disorder, which of the images shows a system with more entropy? Or, if you prefer, which image has more entropy?



**Figure 1:** A crystal and a glass

**Read This:** Now consider entropy as it relates to the number of available microstates for the given macrostate. Think of *crystal.bmp* as a 256 x 256 image of a very expansive crystal. We should consider the macrostate to be “a 256 x 256 image of this crystal.” We can show without too much work that there are only 441 unique 256 x 256 images that look like this crystal. (Why? 21 pixels separate adjacent atoms and  $21 \times 21 = 441$ . Basically, the top left corner of the top left atom could be in any of 441 pixels.) These unique images are the microstates.

**Read This:** With the glass, we could basically put any atom anywhere in the image and still consider it to be a glass because we don’t really care where the atoms are. There are tons of microstates for the glass. It’s on the order of  $10^{509}$ , because there are 65,536 pixels where we can place each of the 169 atoms. The number of ways to do this is  $\binom{65536}{169}$ , read “65,536 choose 169.” This is a very large number.

**Checkpoint 1.2:** Now, let’s start exploring entropy as ignorance. Here’s a challenge. You are looking at the image of the crystal. Your friend cannot see the image. You must relay instructions to your friend such that she can produce an identical copy of the image of the crystal. What would you tell her? Don’t leave anything out. (See Appendix A for a close-up view of *crystal.bmp*.)

**Checkpoint 1.3:** Repeat the exercise in Checkpoint 1.2 for the image representing the glass. This time, however, you can feel free to summarize your instructions. We only have so much time!

**Read This:** You should see that the crystal is a lot easier to describe to your friend. Basically, if you know a few parameters for the crystal (you should have described these parameters in Checkpoint 1.2), you can predict the location of any atom in the crystal. That is, we have little ignorance about the locations of the atoms. That goes hand-in-hand with the crystal having little entropy. In contrast, knowing the location of one atom in the glass gives no clues about the locations of other atoms. We cannot predict the locations of atoms in the glass. The glass, therefore, must have relatively high entropy.

**Read This:** Compression schemes exploit patterns in an image to store files more efficiently. It turns out that the equations that quantify the ability to compress a file are the same equations

used to quantify entropy in thermodynamics! You will do some quantitative analyses in Part IV. Let's stay qualitative for the time being. Just keep the big claim in mind: *files with more entropy are harder to compress*.

**Checkpoint 1.4:** Record the sizes of *crystal.bmp* and *glass.bmp* in bytes. To find the size, control-click the file's icon and select "Get Info" from the drop-down menu. (We want the "Size", not the "Size on Disk".)

**Checkpoint 1.5:** Which of the two images do you suspect your computer can compress into the smaller file? Discuss.

**Do This:** Compress the files. To compress a file, hold control and single-click its icon. The fifth option in the drop-down menu should say "Compress [filename]." If you are successful, a ZIP file should appear in the same folder as the file.

**Checkpoint 1.6:** Record the sizes of the compressed files in bytes. To find the size, control-click the file's icon and select "Get Info" from the drop-down menu. (We want the "Size", not the "Size on Disk".)

**Checkpoint 1.7:** Was the prediction you made in Checkpoint 1.5 correct? If not, discuss where your reasoning failed.

**S1**

**Synthesis Question 1 (15 Points):** Explain whether or not your experiments in Section 1 support the claim that files with more entropy are harder to compress. Be sure to cite your data. Further, make it clear which interpretation of entropy you are using in your analysis.

## *Part II: Compressing Language I – Hangman*

We started our foray into entropy and information with a game in the Pre-Lab. It's time for game #2. Once again, this is not a game to play merely for amusement. You hopefully learned a little bit in Part I about how computers can compress images and how there is a connection to thermodynamics. In the remaining parts of the lab, you will be working with language. You will eventually see that the concepts are all the same, even if this is not immediately obvious.

### **Equipment**

- Pen
- Paper

### **2. Hangman**

Playing hangman is actually a surprisingly useful introduction to the concepts we will cover in Parts III and IV of this lab.

**Checkpoint 2.1:** Play a game of hangman, paying close attention to the strategies you use. (Ask your TA if you are unfamiliar with the game.) A good puzzle should consist of a sentence or phrase. A single word is not a good puzzle. Your group should play multiple games. Each group member should make up at least one puzzle.

**S2**

**Synthesis Question 2 (15 Points):** Write your strategy to solving a hangman puzzle in gruesome detail. You should be able to list **at least six** clearly distinct pieces of strategy. Feel free to write more!

**Read This:** Hangman is fun. It would not be fun if you couldn't answer Synthesis Question 2. Further, the fact that you can write an answer to Synthesis Question 2 has profound implications for compressing a file of English text. The patterns that you exploit as you play hangman are the same patterns that a computer can exploit to compress a file. So let's watch a computer in action!

### *Part III: Compressing Language II – Text Files*

#### **Equipment**

- Computer

#### **3. Saving Text Files**

In this section, you will once again investigate the following claim:

**Files with more entropy are harder to compress.**

This time, though, you will use text files instead of images. Recall that a bitmap uses three bytes for each pixel of an image. Well, a .txt file works in a somewhat analogous way, using one byte to store each character. For example, a .txt file of 4000 characters would require 4000 bytes to store. Like the bitmap, this format is not especially clever and, in most cases, the file can be compressed to a much smaller size.

**Do This:** Open *Word* and type exactly 2000 characters including spaces. (Click on the bottom taskbar where it reads “Words:” to count characters.) The only rules are:

- 1) You can't hit “Return”.
- 2) This must clearly be English (though grammar and spelling mistakes are fine).
- 3) If you don't want to type 2000 characters, you may copy text from the internet. However, after pasting the text into *Word*, delete any spacing between paragraphs. That is, make the text one giant paragraph.

**Do This:** When you reach exactly 2000 characters, save the file as “English.txt” (Select “Save As” from the “File” menu and choose “Plain Text” as the format). Saving with the extension “.doc”

will not work for this experiment. Each group only needs to make one file. You can work together. (To be clear, you are counting *characters with spaces*, not words.)

**Do This:** Make a second text file in *Word* that is exactly 2000 characters including spaces. This time the rules are a little different. The rules are:

- 1) You can't hit "Return".
- 2) This should be complete gibberish! Make this as random as you possibly can. Take turns to make it even more random. (Note: random does not mean "asdfasdfasdf." Just because it makes no sense does not necessarily mean it's random. Peck all over the keyboard.)

**Do This:** When you have exactly 2000 characters with spaces, save this chef d'oeuvre as "random.txt". Once again, using the extension ".doc" will not work.

**Checkpoint 3.1:** By interpreting entropy as disorder, which of the two new files has more entropy?

**Read This:** To be able to relate the multiplicity to the entropy, we must first identify what we mean by the macrostate. We must ask ourselves what macroscopic quantities are associated with the files. The only things we know about the files without looking closely at individual characters are the *language* used in the file and the *length* of the file. Then the macrostate associated with *English.txt* is "2000 characters of English," while the macrostate associated with *random.txt* is "2000 random characters."

**Checkpoint 3.2:** By considering the number of microstates, which of the two files has more entropy?

**Checkpoint 3.3:** Finally, consider the interpretation of entropy as ignorance. Which file has more entropy? You could think about how easy it is to make predictions about the upcoming characters as you peruse the file. Don't forget about that game of hangman that you played!

**Checkpoint 3.4:** Record the sizes of *english.txt* and *random.txt* in bytes. To find the size, control-click the file's icon and select "Get Info" from the drop-down menu. (We want the "Size", not the "Size on Disk".)

**Checkpoint 3.5:** Based on your responses to Checkpoints 3.1 – 3.3, which file do you suspect the computer can compress into the smaller size?

**Do This:** Compress both of your text files using the same procedure you followed with the images.

**Checkpoint 3.6:** Record the sizes of *english.zip* and *random.zip* in bytes. To find the size, control-click the file's icon and select "Get Info" from the drop-down menu. (We want the "Size", not the "Size on Disk".)



**Checkpoint 3.7:** Was the prediction you made in Checkpoint 3.5 correct? If not, explain where your reasoning failed.

**S3**

**Synthesis Question 3 (20 Points):** Explain whether or not your experiments in Section 3 support the claim that files with more entropy are harder to compress. Be sure to cite your data. Further, make it clear which interpretation of entropy you are using in your analysis.

## *Part IV: Compressing Language III – Quantitative Analysis*

Here's where it really comes together. So far, we have made mostly qualitative connections among thermodynamics, entropy, ignorance, information, language, and file compression. Part IV introduces the equations that put a theoretical limit on the compressibility of an image or text. And, as mentioned earlier, they are the same equations we use to quantify entropy in thermodynamics.

### **Equipment**

- Computer
- 1,000,000-character .txt file

### **4. Equations for Entropy**

The entropy,  $S$ , of a macroscopic state is given by

$$S = k_B \ln \Omega \quad \text{Eq. 1} \\ \text{[Moore,T5.5]}$$

where  $k_B$  is a constant and  $\Omega$  is the multiplicity of the macrostate. (The multiplicity is the number of microscopic arrangements that can give rise to the particular macrostate.) In thermodynamics, the constant  $k_B$  is known as Boltzmann's constant, which is chosen so that we can easily connect entropy to temperature.

A key assumption of this equation is that all microstates are equally probable. (This is known as the Fundamental Assumption of Statistical Mechanics.) However, if we are going to quantify the connection between language and entropy, we must get rid of this assumption.

Let's understand why. Consider a string of 20 letters that make some message in English. How many microstates are there? That is, how many possible messages can be sent?

Begin by assuming that all microstates are equally probable (that the first letter is A one twenty-sixth of the time, etc.). There are  $26^{20}$  combinations of 20 letters. Then the entropy, as given by Equation 1, is  $S = k_B \ln 26^{20}$ . (We have ignored spaces and punctuation for simplicity.)

But we began by making a ridiculous assumption: clearly not all combinations of letters are equally probable in an English message! Randomly selecting 20 letters will almost certainly not make any sense,

and it is extremely unlikely that such a message would be sent. English is not random! First, there is a relatively small set of words to choose from. Further, some words are much more common than other words (“the” vs. “gnu”) and some letters are much more common than other letters (“s” vs. “q”).

Therefore, we need a more general entropy equation to deal with systems whose microstates are not equally probable. And here it is:

$$S = -k_B \sum_i p_i \ln p_i \quad \text{Eq. 2}$$

where the sum is over all of the microstates and  $p_i$  is the probability of the  $i$ th microstate. That likely sounds and looks confusing, so let’s pause to make sure this equation makes some sense.

**S4** **Synthesis Question 4 (10 Points):** The claim is that Equation 2 is more general than Equation 1. We should therefore expect that Equation 2 reduces to Equation 1 under some circumstances. Consider a macrostate with  $N$  equally likely microstates. Show that Equation 2 and Equation 1 give the same value for the entropy of the macrostate.

### A Fundamental Limit to Compression

When discussing entropy in the context of thermodynamics, using Boltzmann’s constant and the natural logarithm makes sense due to the connection with temperature. However, to consider the entropy of a message, we can actually pick a different constant and a different logarithm to make things more intuitive. That’s what Claude Shannon did in 1948, when he essentially invented information theory. The equation for the so-called Shannon entropy of a message is given in Equation 3:

$$S = - \sum_i p_i \log_2 p_i \quad \text{Eq. 3}$$

such that the entropy is measured in **bits** rather than J/K. As before, the sum is over all microstates. We could also express this entropy in **bytes** where 1 byte is equal to 8 bits.

Equation 3 happens to be quite remarkable. This is the equation that we can use to put a fundamental limit on our ability to compress a message or a file. In other words, Equation 3 tells us the fundamental minimum number of 0’s and 1’s that we need to store a particular message. Here’s how we use it: if Equation 3 tells us that the entropy of a particular message is 1000 bits, then we cannot expect to compress the message to a size smaller than 1000 bits. We will not prove these claims concerning Equation 3. However, you will put the claims (and your computer) to the test very shortly.

### 5. First Transmission: Coin Flips

To get a feel for how we can use this equation, let’s consider a common scenario when discussing entropy: flipping coins. Let’s say you flip a coin five times and you’d like to send a message to your friend telling him the exact outcome of the flips. You could send a message like “heads, tails, heads, heads, heads.” Or you could be a little cleverer and invent a compression scheme where  $H$  means heads and  $T$  means tails so that the message reads “HTHHH”. Notice that no information is lost. The message is just

more succinct. However, if you used a .txt file to send this message, it would represent both  $H$  and  $T$  as 8 bits each ( $H = 01001000$ ,  $T = 01010100$ ) for a total of 40 bits (or 5 bytes). Doesn't that seem unnecessary? Why not just let 0 mean heads and 1 mean tails? Then the message would be stored as "01000" with no information lost. We have compressed the message down to 5 bits without any loss of information.

Can we do better? To answer that question, you will need to calculate the entropy of the message. We can apply Equation 3 character-by-character to find the entropy of the whole message. To do this, we treat a message as if each character is its own system. Then we add up the entropy of each character to get the entropy of the whole message. (See Equation T5.7 in Moore to motivate this approach.)

**S5**

**Synthesis Question 5 (10 Points):** Find the entropy of a transmission of five coin flips where "heads" is transmitted as 0 and "tails" is transmitted as 1. First, apply Equation 3 to the *first character* of the message to find the entropy of the first character. Here's how to start. There are two possible microstates for the first character (0 and 1), so our sum will contain two terms.

$$S = -p_0 \log_2 p_0 - p_1 \log_2 p_1 \quad \text{Eq. 4}$$

Start by copying that. It's up to you to choose the proper values for the probabilities and simplify the expression. Second, find the total entropy of the message. Just multiply the entropy of the first character by the number of characters in the message. Make sure your calculation is speckled with English to add clarity to your work.

**Read This:** You should have found that the entropy was 5 bits. That means that we've done a very good job at compressing our message. Be aware that it's possible to devise codes that send some results in less than 5 bits, but there is no way to do better than this *on average*. For example, we could let 0 represent "HHHHH" so that once in awhile, our transmission would be very short. But then many other outcomes would require transmissions that are longer than 5 bits.

## 6. Second Transmission: Three-Letter Language

This section is inspired by an exercise in a statistical mechanics text by James P. Sethna<sup>3</sup>. Consider a language that consists of three characters:  $a$ ,  $b$ , and  $c$ . On average,  $a$  and  $b$  are equally likely (25% each) while  $c$  is twice as common (50%). We will call this the Three-Letter Language.

**Checkpoint 6.1:** What is the entropy of a single character of a message in the Three-Letter Language? (You might take inspiration from Synthesis Question 5.)

**Checkpoint 6.2:** How many bits (0's and 1's) are necessary to send a message in the Three-Letter Language that is  $N$  characters long (on average)?

**Checkpoint 6.3:** Devise a compression scheme that will convert a message in this language into 0's and 1's as efficiently as possible. This takes some cleverness. If you are having difficulties, you might want to look to Morse Code for inspiration.

**Checkpoint 6.4:** Test your compression scheme on a short message in the Three-Letter Language. (A short message is a short string of a's, b's, and c's.) Translate the message into 0's and 1's using your code, then make sure you can back-translate into a's, b's, and c's.



**Do This:** If you think you have a scheme that works, check with your TA to make sure that your compression scheme is up to snuff.

**Checkpoint 6.5:** Discuss the success of your compression scheme. For example, did the answer to Checkpoint 6.2 hold true when used on your short message? If not, why not?

**Read This:** Statistical mechanics describes extremely large systems extremely well. Similarly, compression schemes can really show their power when working on large files. With this in mind, the remaining Checkpoints will focus on a message in the Three-Letter Language that is one million letters long.

**Do This:** Go to the In-Lab Links page and find a randomly generated 1,000,000-character-long message. Download it as instructed. This .txt file contains one-million characters. About 25% of them are a's. About 25% are b's. About 50% are c's.

**Checkpoint 6.6:** What is the fundamental limit to which you can expect this message can be compressed? You might want to use your answer to Checkpoint 6.1. Record your answer in bits and bytes. (Recall that there are 8 bits in 1 byte.)

**Do This:** Compress your file containing one million characters.

**Checkpoint 6.7:** What is the size of the compressed file? Record your answer in bits and bytes.

**Checkpoint 6.8:** How close does the computer come to the maximum possible compression? (Make sure this is quantitative.) Could you do better than the computer?



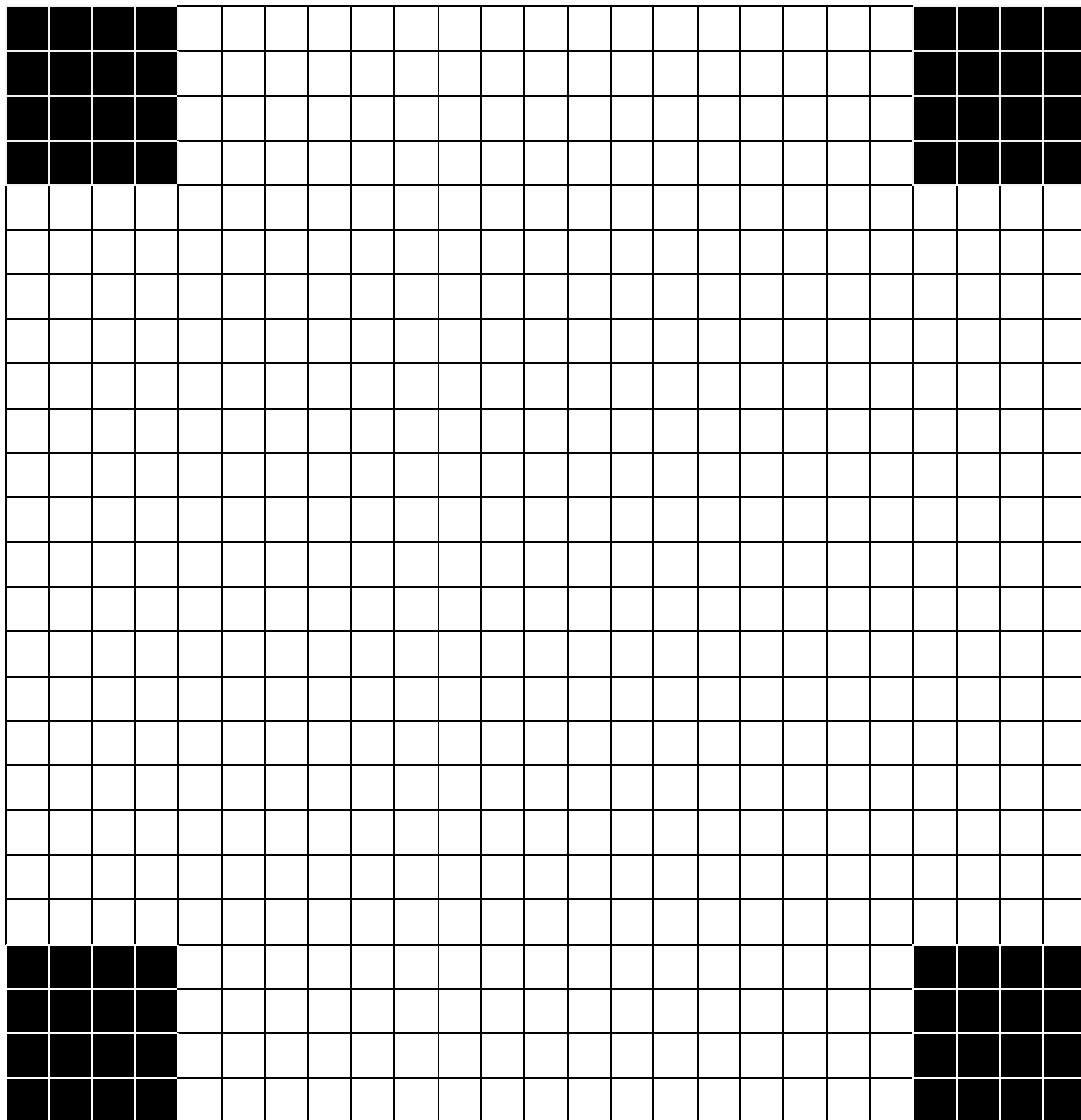
**Synthesis Question 6 (30 Points):** Compare the success of *your* compression scheme and the success of the *computer's* compression scheme to the fundamental limit found using Equation 3. An excellent response should include the following.

- A derivation of the fundamental limit on the size to which you could expect to compress your file with one million characters.
- A description of your compression scheme along with evidence that it works well.
- An evaluation of the computer's ability to compress the file with one million characters.
- A conclusion comparing your ability and the computer's ability to compress a message written in the Three-Letter Language.

## *References*

- [1] Leff, Harvey S., Rex, Andrew F. (1990). *Maxwell's Demon: Entropy, Information, Computing*. Princeton University Press, Princeton, NJ.
- [2] Feynman, Richard P. (1996). *Feynman Lectures on Computation*. Addison-Wesley Publishing Company, Inc., Reading, MA.
- [3] Sethna, James P. (2008). *Entropy, Order Parameters, and Complexity*. Oxford University Press, Inc., New York.

*Appendix A: Close-up of crystal.bmp*



Each square represents a single pixel.