Information entropy questions for Advanced Thermodynamics. #1 and #2 adapted from Sethna's book, while #3 is based on Avinery, Ram, Micha Kornreich, and Roy Beck. "Universal and accessible entropy estimation using a compression algorithm." *Physical review letters* 123, no. 17 (2019): 178102.

1. For a point source of unit mass at the origin, the diffusion equation yields the following evolution of probability density for material distribution in one dimension x with respect to time t:

$$p(x,t) = \frac{1}{\sqrt{4\pi Dt}} e^{-\frac{x^2}{4Dt}}$$

where D is the diffusivity of the material. Use differential entropy formulation to show that the diffusion process leads to a increasing entropy with time.

Solution:

$$S = -k_B \int_{-\infty}^{\infty} p \ln p \, dx$$

$$S = -k_B \int_{-\infty}^{\infty} p \ln \frac{1}{\sqrt{4\pi Dt}} e^{-\frac{x^2}{4Dt}} dx$$

$$S = -k_B \int_{-\infty}^{\infty} p \left(-\frac{1}{2} \ln 4\pi Dt \ - \frac{x^2}{4Dt} \right) dx$$

$$S = \frac{k_B}{2} \ln 4\pi Dt \ \int_{-\infty}^{\infty} p dx + \frac{k_B}{4Dt} \int_{-\infty}^{\infty} p(x^2) dx$$

$$S = \frac{k_B}{2} \ln 4\pi Dt \ \times 1 + \frac{k_B}{4Dt} \times 2Dt$$

$$S = \frac{k_B}{2} (\ln 4\pi Dt \ + 1)$$

$$\frac{dS}{dt} = \frac{k_B}{2t} > 0 \ \forall \ t > 0$$

2. Riffle shuffle involves dividing a deck of cards into two halves (*top* and *bottom*) and then elegantly placing one card from each half randomly in sequence.

Consider the deck after a riffle; each card in the deck either came from the top portion or the bottom portion of the original deck. A riffle shuffle makes each of the 2^{52} patterns *tbbtbttb* . . . (denoting which card came from which portion) equally likely.

It is clear that the pattern *tbbtbttb* . . . determines the final card order: the number of t's tells us how many cards were in the top portion, and then the cards are deposited into the final pile according to the pattern in order bottom to top.

a) What is the information entropy of the deck before it is shuffled? After it is completely randomized?

Before: only one combination →$\boldsymbol{S = 1 \times log_2 1 = 0}$

After: $\boldsymbol{52!\ arrangements = S = log_2(52!)}$

b) Ignoring the possibility that two different riffles could yield the same final sequence of cards, what is the information entropy after one riffle?

$$\boldsymbol{log_2(2^{52})}$$

c) Continuing to ignore the possibility that two different sets of *m* riffles could yield the same final sequence of cards, how many riffles would it take for our upper bound for the entropy to pass that of a completely randomized deck?

$$\boldsymbol{m \times log_2(2^{52}) = log_2(52!)}$$

$$\boldsymbol{m} = \frac{\mathbf{225.58}}{\mathbf{52}} = \mathbf{4.34}$$

→ 5 minimum shuffles

3. Can the unlimited trial subscription of WinRar® give you the ability to find the entropy in a simulation? In this exercise we try to look for such connections:

Earlier in the course it was emphasized that the extent to which a given message can be compressed without loss is related to information (Shannon) entropy. The *.zip compression scheme (available at your familiar Mac or PC) is one such lossless compression scheme and is known to get simple files close to the Shannon limit for their file size.

During my simulations I generate a lot of data. Philosophically, a given simulation file contains everything there is to the system, waiting to be found out. One of the harder to determine quantities is the thermodynamic entropy. Could I somehow just compress one of my simulation data files and try to obtain an estimate for the real entropy?

For simplicity, consider a four-state system in a canonical ensemble: say there is this hypothetical particle that can occupy four distinct energy levels: 1ϵ, 2ϵ, 70ϵ, and 80ϵ. Let's call these states 1,2,3, and 4 respectively.

1. Write the analytical expressions for the system at a given T for:
    a. Probabilities of each state: p_1, p_2, p_3, p_4
    b. Canonical partition function
    c. Thermodynamic entropy
2. Suppose I simulate this system by randomly generating numbers {1,2,3,4} in the proportion of their probabilities at a given temperature T. At the end I obtain a data file which looks like:

```
3,4,3,2,4,4,2,3,2,4,3,2,4,2,4,4,1,2,1,2,1,3,3,4,4,1,4,3,3,2,1,1,3,4,2,1,2,1,4,2,3,4,2,1,4,
4,3,3,4,3,1,3,1,2,3,2,1,4,4,1,1,1,4,3,1,2,1,2,2,4,3,4,2,3,3,2,2,2,4,4,1,4,3,2,2,2,3,3,2,4,
1,2,2,4,3,4,4,1,3,4,3,2,3,1,1,4,2,2,4,3,2,2,2,3,2,2,1,3,2,1,4,4,4,1,1,1,4,4,4,1,3,2,3,3,1,
2,1,3,4,2,2,3,3,1,3,3,3,4,4,2,2,2,1,3,4,3,1,4,2,4,4,4,2,2,1,3,3,2,3,3,2,4,2,2,2,4,4,1,2,1,
3,2,2,4,3,4,1,2,4,3,1,3,1,2,4,2,1,1,1,3,3,4,2,3,3,1,4,1,3,3,3,2,3,2,4,4,2,4,3,4,3,1,3,4,4,
1,3,2,3,4,3,1,2,1,2,1,1,1,2,3,1,2,2,4,2,4,1,3,3,2,4,3,2,4,3,1,4,3,1,2,1,1,3,1,2,3,3,4,4,3,
1,1,3,1,2,4,4,4,2,2,3,1,1,1,2,4,3,4,3,3,2,2,2,3,4,3,1,1,1,2,3,2,3,3,1,2,3,1,2,3,4,3,4,1...
```

If I zipped this file with N such digits stored in it, the compression algorithm tries to compress these digits, eventually leading to the minimum amount of information contained in the data. The hypothesis is that the size of this zipped file should relate to the information entropy in this data which, we conjecture, relates to the actual entropy of the system at that temperature T.

However, we would need to set the baseline and scaling for the entropy, so that we can compare our results with 1(c). The compression algorithms usually have some headers and other auxiliary data written onto the final file, which generally contributes the same amount of overhead. To account for this we will obtain the extreme case of all N digits being 1's (or any digit for that matter), [1,1,1,1,1,1,1,1,1….] and compress it. Let this file be compressed to size b_l bytes. Theoretically, this is the least entropy state. Now if a given data file with N digits is compressed to a size b bytes, then the actual difference of the information entropy $S_I \propto (b - b_l)$. We can also generate a maximum entropy state data file by generating N random digits out of the set {1,2,3,4}. Let the compressed size of this file be b_m. Then we can obtain a scaling from the least entropy state to the maximum entropy state as:

$$\eta = \frac{b - b_l}{b_m - b_l}$$

The value for η will go from zero to one as the system goes from a state of least entropy to the state of maximum entropy.

a. At T={0.1, 1.0, 10, 100, 1000, 10000}, generate data files that have $N = 10^7$ instances of states in proportion with the analytical probabilities.

b. Generate baseline and maximum entropy states, obtain the values of b_l and b_m.

c. Use compression to obtain $\eta(T)$ from the files generated in part 2(a). Compare this with the analytic values you can obtain for η from your result in part 1(c). Use error bars. Feel free to generate data points for other intermediate temperatures. Attempt to explain any differences/similarities.

3. Would this approach be suitable for a file from the results for the Monte Carlo simulation of the tetramer model from Problem 2 in PS5? Say you were given a file with the four configurations {I, L, S, and U} mapped to {1,2,3, and 4} and the result is a datafile with a string of these numbers. Would a compression approach result in a reliable estimate of entropy of the system? If yes, argue why (feel free to use analytical arguments). If no, under what circumstance will it work? *(Only comment on the validity of the approach, the actual computation is not necessary)*.

Notes on file handling in Matlab:

Any array *d* containing numbers can be written into a data file using the following command:

```
dlmwrite('d',data,'delimiter',',')
```

The file can be zipped to another file *'d.zip'* using the following command:

```
zip('d','d')
```

The size of such a file can be obtained with the following sets of command

```
f=dir('d.zip') %opens the file object
b=f.bytes %b is your file size
```

After use, the files can be deleted :

```
delete('d')
delete('d.zip')
```