1. Specifically, answer the following under the O(*NT*) assumption:

   If both N and T are doubled, the empirical runtime of BaseMarkov would be around 4 times its original runtime because 2N(2T) = 4NT.

   If N is much larger than T and T is doubled, the empirical runtime of BaseMarkov would be around 2 times its original runtime because N(2T) = 2NT. Similarly, if N is much larger than T and N is doubled, 2N(T) = 2NT.

2. Specifically, answer the following under the O(*N* + *T*) assumption:

   If both N and T are doubled, the empirical runtime of EfficientMarkov would be around 2 times its original runtime because 2N + 2T = 2(N + T).

   If N is much larger than T and T is doubled, the empirical runtime of EfficientMarkov would not significantly change, as N is the main determinant of the runtime. Conversely, if N is much larger than T and N is doubled, the empirical runtime would almost double.

3. Explain whether the timings presented in the example provide evidence supporting the characterization of the runtime complexity using BaseMarkov as O(*NT*). Reference the actual timings in the example, as well as the hypotheses you made in questions 1 and 2.

   The timings presented in the example indicate that O(*NT*) is not completely accurate as a measure of the runtime complexity. When *N* is held constant and *T* is increased, the runtime doubles each time *T* doubles, which is to be expected of a runtime complexity of O(*NT*). However, when *N* is increased and *T* is held constant, the runtime does not double as *N* doubles, rather, it increases at a seemingly constant rate of around 0.4 seconds. If the runtime complexity of BaseMarkov was O(*NT*), we would expect that no matter which variable *N* or *T* is doubling, the runtime would double because 2N(T) = N(2T) = 2NT.

4. ***Report your timings.*** Explain whether the timings you report provide evidence supporting the characterization of the runtime complexity using EfficientMarkov as O(*N*+*T*). Reference the actual timings you report, as well as the hypotheses you made in questions 1 and 2.

```
time source #chars
0.092 487614 1000
0.118 487614 2000
0.125 487614 4000
0.125 487614 8000
0.084 487614 16000
0.085 487614 32000
0.098 487614 64000
```

```
0.078 487614  4096
0.164 975228  4096
0.301 1462842 4096
0.361 1950456 4096
0.461 2438070 4096
0.595 2925684 4096
0.649 3413298 4096
0.762 3900912 4096
0.884 4388526 4096
0.995 4876140 4096
```

In the first set of timings, *N* (source) is much larger than *T*, and as *T* doubles, the empirical runtime does not seem to be affected by this change. The runtimes for character counts of `16000`, `32000`, and `64000` are all lower than those of `2000`, `4000`, and `8000`. As *N* is doubled and *T* is kept constant, the runtime increases at a seemingly constant rate of less than 0.01. Both of these indicate that the runtime of EfficientMarkov is O(*N* + *T*) because we would not expect the runtime to change significantly as *T* changes if *N* is much larger, and we would expect the runtime to change as *N* changes.

5. Do you think new research code in AI/ML should be more open source? Why, or why not?

   I believe that AI/ML researchers have a responsibility to share their processes with the public due to the possibility of implicit biases that may affect the way a program functions. For example, transparency in the development process for the Google photos categorization program could have aided in its ability to account for inequities in the training process. Without transparency, instances like this will continue to occur until everyone is adequately represented in an AI's training process.

   The obstacle to doing so is financial, as research in AI/ML is extremely costly, and those most able to fund such research generally prioritize profit over making their programs publicly accessible. The power lies in the hands of large tech companies and their willingness to compromise profit margins to benefit a greater total population.