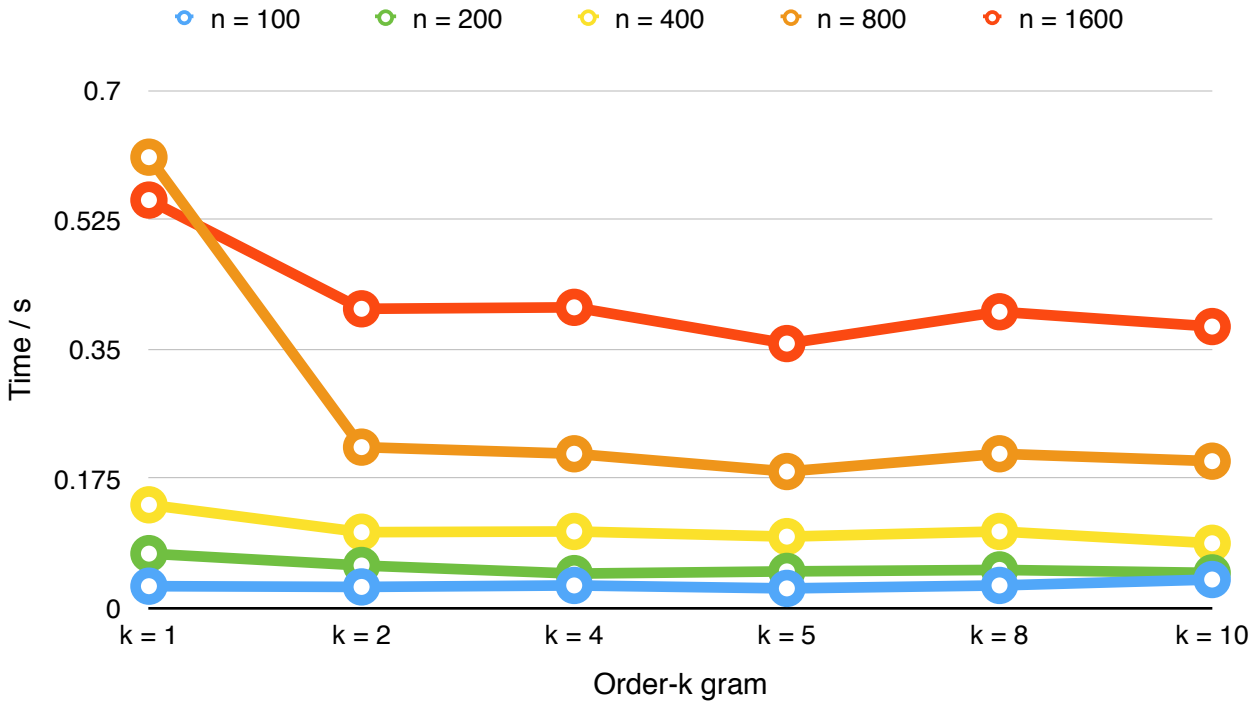
Question 1.

How long does it take using the provided, brute force code to generate order-5 text using Romeo and Juliet (romeo.txt) and generating 100, 200, 400, 800, and 1600 random characters. Do these timings change substantially when using order-1 or order-10 Markov models? Why?

Time to generate for order-k gram with Romeo and Juliet source text

Order-k gram for n-th characters (Averaged over 10 runs)	k = 1	k = 2	k = 4	k = 5	k = 8	k = 10
n = 100	0.03	0.029	0.031	0.027	0.031	0.039
n = 200	0.074	0.058	0.047	0.05	0.052	0.048
n = 400	0.14	0.103	0.104	0.097	0.104	0.088
n = 800	0.61	0.218	0.209	0.185	0.209	0.199
n = 1600	0.552	0.405	0.407	0.358	0.401	0.381



- For order-k grams, increase in k leads to shorter times.
- This could be because for a smaller value of k, there are a larger number of such order-k grams. Having to allocate memory to more order-k grams would increase timings.
- From an initial value of k = 1 to k = 2, the size of order-k gram for values of k fall steeply.
- From k = 2 onwards, the timings stay generally similar.

Question 2.

Romeo has roughly 153,000 characters. Hawthorne's Scarlet Letter contains roughly 500,000 characters. How long do you expect the brute force code to take to generate order-5 text when trained on hathorne.txt given the timings you observe for romeo when generating 400, 800, 1600 random characters? Do empirical results match what you think? How long do you think it will take to generate 1600 random characters using an order-5 Markov model when the King James Bible is used as the training text — our online copy of this text contains roughly 4.4 million characters. Justify your answer — don't test empirically, use reasoning.

Time to generate for order-k gram with Romeo and Juliet source text

Order-k gram for n-th characters (Averaged over 10 runs)	k = 1	k = 2	k = 4	k = 8
n = 100	0.03	0.029	0.031	0.031
n = 200	0.074	0.058	0.047	0.052
n = 400	0.14	0.103	0.104	0.104
n = 800	0.61	0.218	0.209	0.209
n = 1600	0.552	0.405	0.407	0.401

Time to generate for order-k gram with Scarlet Letter source text

Order-k gram for n-th characters (Averaged over 10 runs)	k = 1	k = 2	k = 4	k = 8
n = 100	0.093	0.079	0.069	0.067
n = 200	0.186	0.147	0.143	0.139
n = 400	0.371	0.282	0.271	0.277
n = 800	0.775	0.636	0.552	0.511
n = 1600	1.52	1.154	1.106	1.078

- Given Hawthorne's Scarlet Letter is 500,000 characters to Romeo's 153,000 characters (roughly 3.5x as large), one would expect a 3.5x increase in timings for Scarlet Letter.
- As shown in the two tables above, we see a general pattern that matches our prediction: the relative increase in the number of characters is matched by the timings of order-k grams.
- While we would expect that the timings to increase proportionately with the the same magnitude of increase in the size of the source text, we witness increasing savings in timings as the source text gets larger.
- I would guess that there is a minimum overhead required regardless of the size of the source text and subsequent increases in the size result in timing increases that are smaller in magnitude.

Question 3

Provide timings using your Map/Smart model for both creating the map and generating 200, 400, 800, and 1600 character random texts with an order-5 Model and romeo.txt. Provide some explanation for the timings you observe.

Time to generate for order-5 gram with different models

Order-k gram for n-th characters (Averaged over 10 runs)	Brute-force method	Map-Markov method
n = 100	0.028	0.0010
n = 200	0.07	0.0010
n = 400	0.094	0.020
n = 800	0.196	0.030
n = 1600	0.483	0.050

- We see incredible time savings with the Map Markov model in comparison to the brute-force method.
- The savings are often greater than a order of magnitude, see t = 0.07s v. t = 0.0010 (for n = 200) and t = 0.196s v. t = 0.03 (for n = 800).
- For the Map-Markov model, the time taken to create the map is not taken into account. The map is only created once and is only recreated for a new value of k.
- Because the map is already created, it is much for efficient to access and retrieve the data from an available data source.
- Contrast this to the brute-force method, which had to generate the data order-k gram each time by iterating over the source text each time.