The purpose of this assignment is to give you some experience building a tokenizer, and some practice thinking about the kind of math we'll use in the course.

Problem 1 (2 points) *Content from lecture 2*
Your trained tokenizer's vocabulary will typically contain both the token `the` (no leading space) and the token `␣the` (with a leading space). These are independent vocabulary items with independent token IDs and, once we train a language model, independent embeddings.

(a) Give one example context in which each token occurs, and say briefly how their learned embeddings might differ.

(b) A user prompts a trained language model with `"My favorite animal is the␣"` (note the trailing space) instead of `"My favorite animal is the"`. Using part (a), explain why the trailing space can noticeably degrade the model's next-word prediction.

Problem 2 (2 points) *Content from lecture 2*
In Lecture 2, the encoding algorithm for a subword vocabulary was described as greedily and iteratively compute the longest prefix of the remaining text that is an element of the vocabulary, emit that token, remove the prefix, and continue until the text is empty. The lecture notes that this does not guarantee the shortest possible encoding, and asks why. Consider the vocabulary

$$\mathcal{V} = \{\texttt{a},\ \texttt{b},\ \texttt{c},\ \texttt{d},\ \texttt{ab},\ \texttt{bcd}\}$$

and the input string `abcd`.

(a) Write out the token sequence produced by the greedy algorithm, then exhibit a valid encoding of `abcd` under $\mathcal{V}$ that uses fewer tokens.

(b) In one or two sentences, describe the general failure mode of the greedy algorithm that your example illustrates.

Problem 3 (5 points) *Content from lecture 1 and 3 (background)*
Demonstrate that the gradient of the softmax function with respect to an input index that receives (approaching) zero probability is (approaching) zero. Here's some notation to use: let $x \in \mathbb{R}^d$. For some fixed j, you're given that the output of $\text{softmax}(x)_j$ is very close to zero. Compute the gradient of the function. Then show that when $\text{softmax}(x)_j$ is close to zero, the gradient $\nabla_{x_j}\text{softmax}(x)_j$ is also close to zero. Formalize this closeness as you wish. Recall that

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{d}\exp(x_j)} \tag{1}$$

Problem 4 (5 points) *Content from lecture 1 and 3 (background)*
In lecture, we motivated the language modeling objective by saying we'd like the average probability we assign to texts drawn from a distribution D to be high:

$$\mathbb{E}_{(w_1,\ldots,w_k)\sim D}\left[P(w_1,\ldots,w_k)\right]$$

We then said that we'll instead work with the log-probability, "for various reasons including numerical stability," arriving at our learning objective:

$$\text{Minimize}\quad \mathbb{E}_{(w_1,\ldots,w_k)\sim D}\left[-\log P(w_1,\ldots,w_k)\right]$$

Since the log is monotonic, it may seem like these two objectives must lead to the same learned model.[1] Let's check. Consider a data distribution over just two strings s_1, s_2,

[1] Monotonicity does guarantee that $\log\mathbb{E}[P]$ has the same maximizer as $\mathbb{E}[P]$.

with

$$D(s_1) = 0.9, \qquad D(s_2) = 0.1,$$

and suppose our model is free to assign any distribution over the two strings: $P(s_1) = p$, $P(s_2) = 1 - p$, with $p \in [0, 1]$.

(a) Find the value of p that maximizes $\mathbb{E}_{w \sim D}[P(w)]$.

(b) Find the value of p that maximizes $\mathbb{E}_{w \sim D}[\log P(w)]$.

(c) In one or two sentences, explain why the two objectives disagree. What is it about the shape of log that keeps the optimum in (b) away from $p = 1$?

Problem 5 (10 points) *Content from lecture 2*
Complete the tokenizer code provided in `a1.ipynb`, and upload to gradescope.

Problem 6 (1 points) *Content from lecture 2*
In the tokenizer you trained, when you tokenize the snowman emoji, in the line marked `Individual 'decoded' token strings:`, what do the snowman tokens look like? Why?