Problem statement:

1. Input 1: A misspelled string
2. Input 2: A list of words in the dictionary
3. Levenshtein distance – the number of “edits” it will take to change a misspelled word into the correctly spelled word.
4. Edit – Insertion, Deletion, Substitution
5. Write a program to calculate the minimum number of edits needed to transform one word to another.

How will I do it?

1. I will find how two words differ.
2. For ex: Sunday and Saturday
3. Calculate the length
   1. Sunday’s length = 6
   2. Saturday’s length = 8
4. This means if length is equal, we can rule out substitution, but can we
   1. Sunday and Senday
   2. The minimum edit would mean
      1. Replace e with u
      2. So, minimum edits = 1.
   3. Therefore, we can’t rule out substitution.
5. Let’s start with brute force
   1. Iterate both strings and find the first difference in the alphabet.
   2. When first letter differs, we can either do
      1. Insertion
      2. Substitution
      3. Deletion
   3. But how would be know which is the minimum edit path
   4. I got it! Since we are making choices, dynamic programming make sense.
   5. The keyword is CHOICES
   6. So, we’ll make a choice and then we’ll calculate which choice leads to least number of edits, meaning we’ll need to store the number of edits
6. Since we don’t need to know which type of “edit” is happening, only that it is happening, we can forget just store the number of edits.