Ethan Nguyen
ethantn04@csu.fullerton.edu
CPSC 335 Algorithm Engineering
Project 2 - Algorithm 2

Project Report - Algorithm 2

Pseudocode

We have a string S of n characters that are lower-case letters and spaces.
We aim to output another string C which is a compressed string that runs k repetitions of x and is replaced with “kx”. For example, we have Heloooo and we aim to output Hel4o. Our k in this instance is 4 and x is o.
If S is empty, the program will return an empty string.
The string C will be initialized.
The program will iterate through a string and for each character from a certain index, it will increment the count which will be representative of the variable k.
We will be using an if, else statement to accomplish this.

```
If S[i] is equal to S[i-1]:
    increment count
else:
    if count > 1:
        append count and S[i-1] to C
    else:
        append S[i-1] to C
        reset count = 1
```

Once the loop is finished, the program will append the last character and count.
Once the appending is done, the program will return C, the compressed string.

Sample input:

```
   string test1 = "ddd";
   string test2 = "heloooooooo there";
   string test3 = "choosemeeky and tuition-free";
```

Sample output:

```
Encoded 1: 3d
Encoded 2: hel8o there
Encoded 3: ch2osem2eky and tuition-fr2e
```

Mathematical Analysis & Efficiency Class

Let n be the length of the input string which is S.
Since it is necessary to loop through the string which consists of n characters, the complexity can be inferred to be O(n).
When appending the string, the complexity is also O(n).
We don’t have any nested loops considering we don’t have for, while statements so there’s no increase in complexity.

Because there are no further indications of an increase of complexity considering this program iterates and reads through strings, the overall complexity is O(n).