Project 2: Empirical Analysis

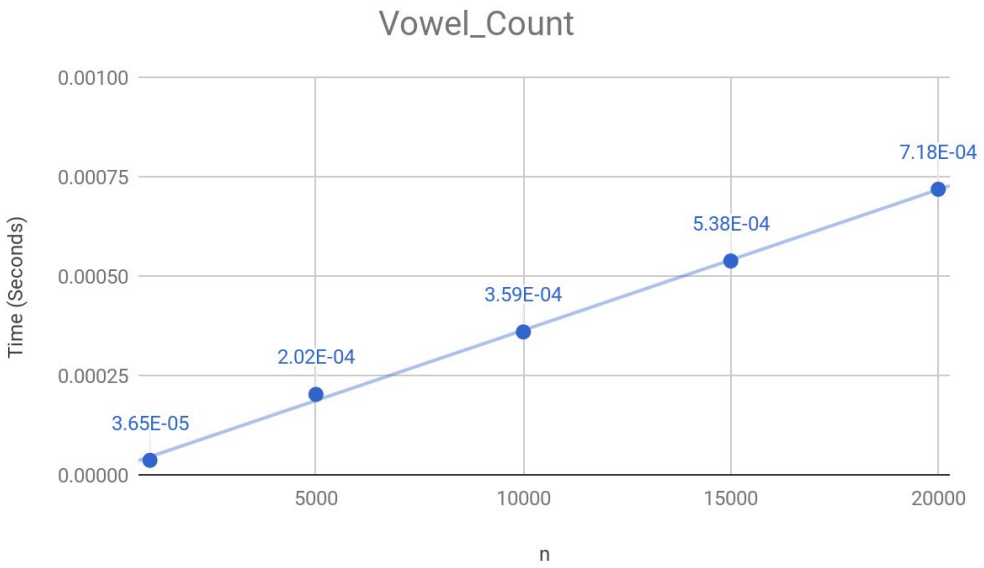
Team Members:

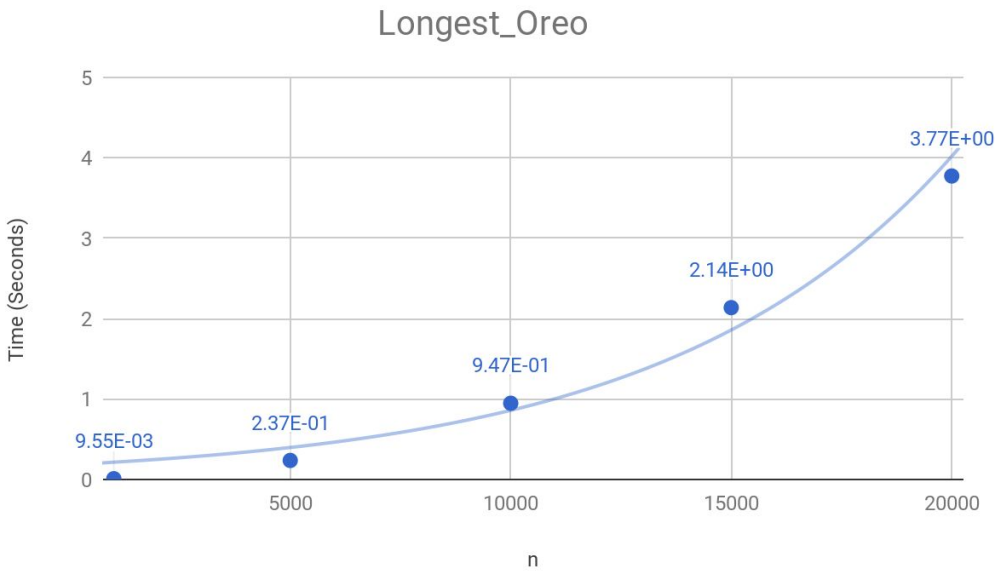
William Bui wqbui3@csu.fullerton.edu
Jasper Chen jsprchn2@csu.fullerton.edu
Alex Ma alexjmma@csu.fullerton.edu

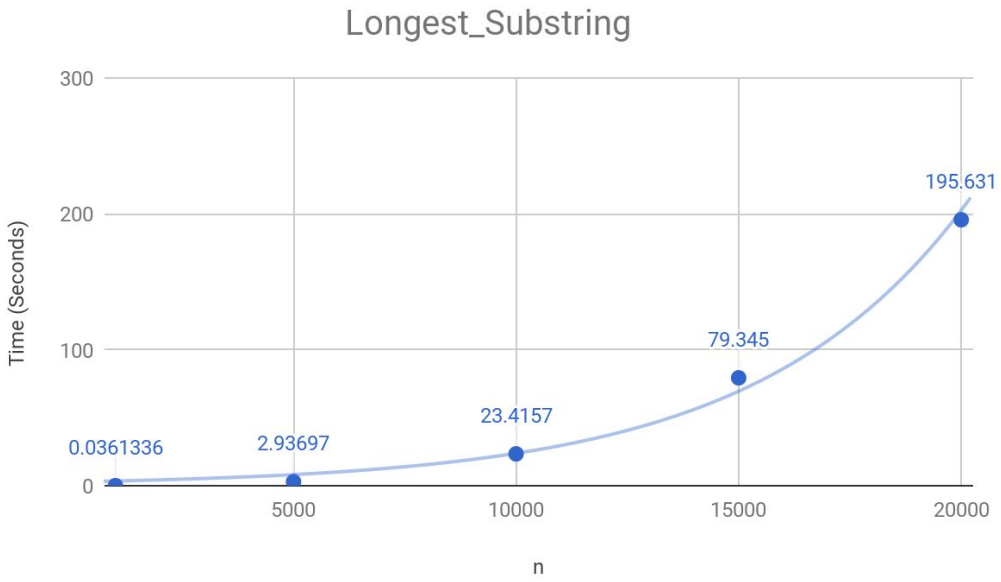
Hypothesis

For large values of n, the mathematically-derived efficiency class of an algorithm accurately predicts the observed running time of an implementation of that algorithm.

Graphs







Questions

1. Provide pseudocode for your three algorithms.

Vowel count

```
vowelCount(string S):
       if (S is empty)
          return 0
       vowelCount = 0
       for x in S
          if (x == Any vowel)
              vowelCount++
       return vowelcount
```

Longest Oreo

```
longestOreo(string S):
   if (s.length() < 2)
      return s
   string1 = ""
   for x in S
      for y from (x+1) in S
         if y == x
            int length = y - x; // length should rep # of spaces between x and y
                if length > string1.length()
                   string1 = str.substr(x, length)
   return string1
```

Longest Repeated Substring

```
longestRepeatedSubstring(string S):
        string LR = " "
        string temp = " "
        string temp2 = " "
        size_t found
        int n = s.length()

        for X in S
            for J in S
                temp2 = s[ J + 1 ]
                found = s.find( temp, J )
                if found != npos
                        if temp.length() > LR.length()
                                LR = temp
                        temp.append(temp2)
                temp = " "        //reset temp
        return LR
```

2. What is the efficiency class of each of your algorithms, according to your own mathematical analysis? (You are not required to include all your math work, ***just state the classes you derived and proved.)***

   Vowel Count
   $T(n) = 2n + 4$
   $O(n)$

   Longest Oreo
   $T(n) = 4n^2 + 4$
   $O(n^2)$

   Longest repeated substring
   $T(n) = 5n^2 + 5$
   $O(n^2)$

3. Is there a noticeable difference in the running speed of the three algorithms? Which is faster, and by how much? Does this surprise you?

   There is a noticeable difference in running speeds. It is most apparent between vowel count and longest oreo/substring. Since vowel count is a relatively simple

algorithm it takes less time to compute and output a value. The time would always be linear because of the efficiency of O(n). The same goes for O(n^2) where it increases quadratically. This comes as no surprise and was seen through our numerous outputs.

4. Are the fit lines on your scatter plots consistent with these efficiency classes? Justify your answer.

   The fit lines are consistent with the efficiency classes. For the vowel count the efficiency class is O(n) and therefore is linear. As shown on the graph, the fit line is linear. For longest oreo and longest repeated substring we can see that the fit line is quadratic with a O(n^2) efficiency. Further proof is that the n values output all increase quadratically and the same goes for O(n) where the output of n is linear.

5. Is this evidence consistent or inconsistent with the hypothesis stated on the first page? Justify your answer.

   The evidence we have gathered shows that it is consistent with our hypothesis. We justified this by running numerous tests until we got the average time for each output for a n value. We then put it in a chart and made a graph from it. From the chart we noticed a pattern that had a strong correlation with our Big-O efficiency. For example if it was O(n) it would be a linear correlation and if it was O(n^2) it would increase quadratically for each n value. Therefore we concluded that our evidence is consistent with the stated hypothesis.