A3 Lab Report

September 25th, 2017

Omar Barazanji

1. **Problem**
   1. The class ProvidedClass in provided.jar contains a method called methodToTime. This method takes an int value like so: methodToTime(int N). Using the constructor of ProvidedClass, which is: ProvidedClass(int key), where the key is my student ID: 903862963, a new object in methodToTime will be created with a unique time complexity.
   2. This unique time complexity’s “big-Oh” (upper bound O(Nk)) needs to be determined empirically, or through observation and experimentation.
   3. Given:
      1. For any key used, the associated time complexity will be proportional to Nk.
      2. This is the property of polynomial time complexity functions T(N):
         1. T(N) α Nk which yields (T(2N) / T(N)) α 2k.
         2. R = T(2N) / T(N) = final time / initial time
         3. k = log2(R)
2. **Procedure**
   1. To determine the big-Oh of the object’s time complexity created in methodToTime, a table must be created that lists the N (problem size), Time, R, and k.
   2. The table will be listed until a reasonable convergence of k occurs.
   3. This convergence is what the big-Oh will be.
      1. For example, if k is approaching ~2, then the big-Oh would be O(N2).
   4. After the big-Oh has been approximated (hypothesized), we will let the more of the code run and verify if our big-Oh approximation was correct or incorrect.
3. **Data Collection, part 1**
   1. Table 1: Running-time data and calculations of myClient

|  |  |  |  |
| --- | --- | --- | --- |
| N | Time (s) | R | k |
| 2 | 0.028844494 | - | - |
| 4 | 0.203747636 | 7.063657834 | 2.82041546 |
| 8 | 2.362981948 | 11.59759197 | 3.53573382 |
| 16 | 37.24632435 | 15.76242439 | 3.98417545 |
| 32 | 635.631364164 | 17.06561321 | 4.09302035 |

1. **Data Analysis**
   1. As seen in the list of k values, it appears the k values are approaching ~4.
2. **Hypothesis**
   1. Since the k values seem to be approaching ~4, then it can be said that the rest of the problem sizes up until 1024 will have a time complexity big-Oh of O(N4).
3. **Data Collection, part 2**
   1. Table 2: Running-time data and calculations of myClient (continued)

|  |  |  |  |
| --- | --- | --- | --- |
| N | Time (s) | R | k |
| 2 | 0.028844494 | - | - |
| 4 | 0.203747636 | 7.063657834 | 2.82041546 |
| 8 | 2.362981948 | 11.59759197 | 3.53573382 |
| 16 | 37.24632435 | 15.76242439 | 3.98417545 |
| 32 | 635.631364164 | 17.06561321 | 4.09302035 |
| 64 | 11322.85774019 | 17.81356047 | 4.154903997 |

1. **Conclusion**
   1. It can be concluded that k is approaching ~4 after waiting for the next N (64).
   2. The big-Oh is O(N4).