**Assignment One**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input Size | Algorithm One (\*103) | Algorithm Two (\*103) | Algorithm Three (\*103) | Algorithm Four (\*103) |
| 2,000 | 9280.6 | 11228.1 | 57.6 | 4191.3 |
| 4,000 | 29633.8 | 33428.7 | 263.2 | 10996.7 |
| 6,000 | 35846.3 | 80012.9 | 557.2 | 13952.0 |
| 8,000 | 37976.9 | 162972.1 | 961.9 | 15252.1 |
| 10,000 | 55775.6 | 246660.2 | 991.8 | 15583.4 |

**Observation and Conclusion**

The graph shows that Algorithm Two (using nested loops to find the largest distance between any two even integers in an array, without creating a new array) is the worst of the four in terms of running time. On the other hand, Algorithm Three (uses a single loop to select even integers from the given array and compute max-min) persists to be the best of the four algorithms with the shortest running time. Algorithm Four (uses streams to filter even integers from the given array and find the maximum and minimum, returning their difference) is observed to be the second best algorithm of the four. Algorithm One is the third best algorithm for the given problem. It creates a new array that is entirely made up of even integers and uses nested loops to find the maximum distance. Algorithm One and Algorithm Two are have similar running time for smaller input. After this value, the running time of Algorithm Two grows a lot faster than that of Algorithm One. It can be concluded that using a single loop without creating a new array is the fastest while using nested loops to on an initially mixed (odd and even) array of integers is the slowest to solve the largest distance problem.

**Question 2**

Prove

Fibonacci : 1, 1, 2, 3, 5…

**Solution**

1. Take two base cases and prove it is true for the two base cases, i.e n = 5 and n = 6

**Show true for**

**Show true for**

1. Assume it is true for

For this assumption to be true both must be true.

1. Prove for

From our assumption in (2) we know that

Expand and simplify the LHS

Since