```
SearchValue,Binary,Jump,Interpolation,Ternary,Exponential,Fibonacci
11965,3100,11100,572200,3700,8000,3100
8024,1700,4200,69100,1400,2500,2000
9501,2500,5300,55300,1400,3200,2800
7658,2300,3900,40200,1400,2700,1700
1017,1900,9000,6300,1600,2800,1900
10148,3700,8900,49100,2100,3300,2300
5365,2800,5900,28700,2000,3900,2700
5089,3700,4800,23500,1300,2700,2000
18006,4000,14300,120400,2000,3000,1900
13647,2200,7200,61800,1500,2500,2600
Averages,2790,7460,102660,1840,3460,2300
```

From my tests, Ternary Search performed the best, followed by Fibonacci, Binary, Exponential, Jump, and finally Interpolation Search.

Ternary search demonstrated its O(logn 3) complexity, barely edging out Binary, Fibonacci and Exponential Searches with their O(logn 2) Complexity. With a space complexity of O(1), this common denominator did not play a part in determining a better performing search.
However, Jump Search's O(√n) Complexity was apparent, taking twice as long as the common Searches. Finally, Interpolation Search's O(n) complexity results in a horrendous time of 102660 nanoseconds, 55x slower than the fastest Ternary Search.