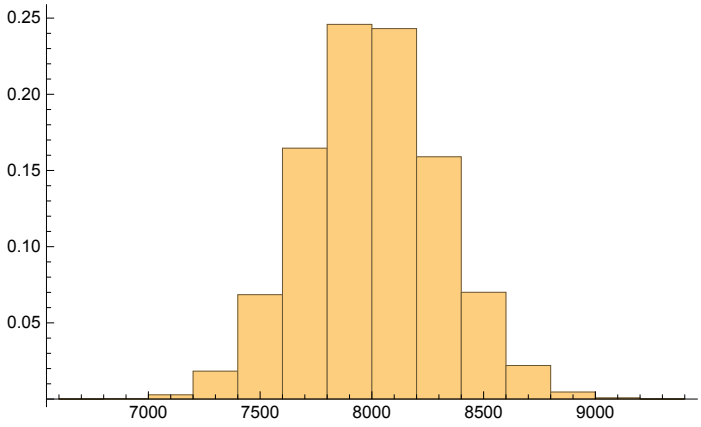
In[1]:=
```
(*3x+1 function *)
f[x_] := If[Mod[x, 2] == 0, x / 2, 3 x + 1];
g[x_] := Module[{},
   (* input is an odd number *)
   temp = 3 x + 1;
   While[Mod[temp, 2] == 0, temp = temp / 2];
   Return[temp];
  ];
```

In[3]:=
```
iteratef[start_, print_] := Module[{},
   count = 0;
   current = start;
   Print["We are starting at ", current, "."];
   While[current > 1,
    {
     current = f[current];
     count = count + 1;
     If[print == 1, Print[current]];
    }]; (* end of while loop *)
  ];
```

In[4]:=
```
iteratefrange[numdo_, startpower_, function_] := Module[{},
   results = {};
   h[y_] := If[function == 1, f[y], g[y]];
   For[n = 1, n ≤ numdo, n++,
    {
     count = 0;
     current = RandomInteger[10^startpower];
     While[current > 1,
      {
       current = h[current];
       count = count + 1;
      }]; (* end of while loop *)
     results = AppendTo[results, count];
    }]; (* end of n loop *)
   Print["Doing ", numdo, "  runs starting at 10^", startpower];
   Print["Average is ", 1.0 Mean[results]];
   Print["StDev is ", 1.0 StandardDeviation[results]];
   Print[Histogram[results, Automatic, "Probability"]];
  ];

Timing[iteratefrange[100000, 1000, 2]]
```

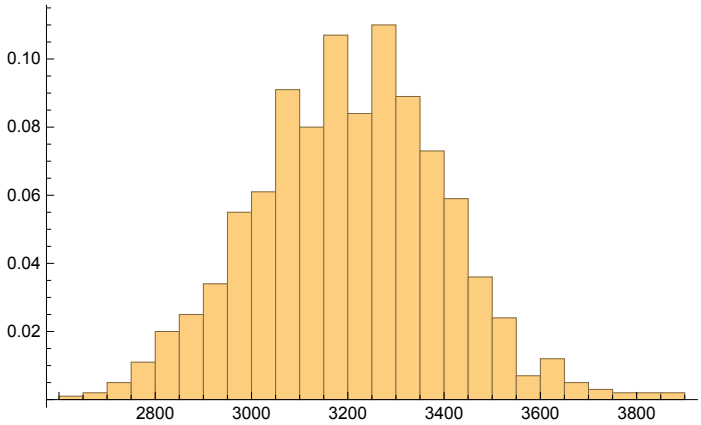
```
Doing 100 000  runs starting at 10^1000

Average is 8003.02

StDev is 304.15
```




```
{6545.02, Null}
```

In[5]:=
```
Timing[iteratefrange[1000, 400, 2]]
```

```
Doing 1000  runs starting at 10^400

Average is 3203.2

StDev is 196.651
```

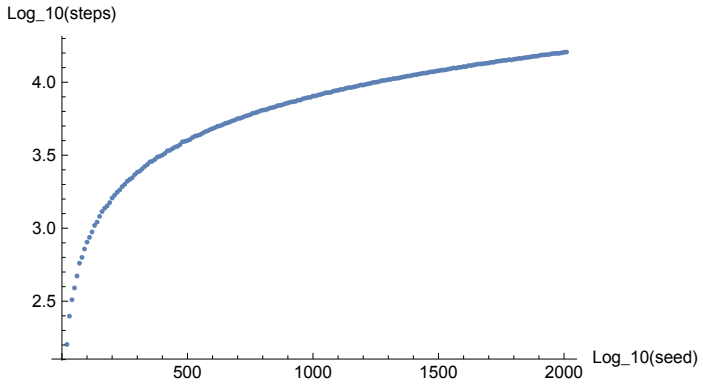


Out[5]=
```
{25.1942, Null}
```

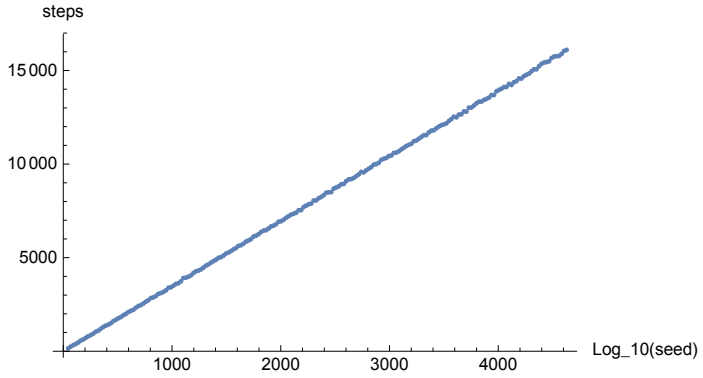
In[11]:=
```
iteratefrangeanalyze[numdo_, startpower_, numpowers_, function_] := Module[{},
   results = {};
   powerresults = {};
   semilogpowerresults = {};
   loglogpowerresults = {};
   h[y_] := If[function == 1, f[y], g[y]];
   For[p = 1, p ≤ numpowers, p++,
    {
     exponent = startpower + 10 * p;
     count = 0;
     If[Mod[p, numpowers / 10] == 0, Print["We have done ", 100 * p / numpowers, "%."]];
     For[n = 1, n ≤ numdo, n++,
      {
       current = RandomInteger[10^exponent];
       While[current > 1,
        {
         current = h[current];
         count = count + 1;
        }]; (* end of while loop *)
       results = AppendTo[results, count];
      }]; (* end of n loop *)
     loglogpowerresults = AppendTo[loglogpowerresults,
       {Log[10., 10.^exponent], Log[10., 1.0 count / numdo]}];
     semilogpowerresults = AppendTo[semilogpowerresults,
       {Log[10.^exponent], 1.0 count / numdo}];
    }]; (* end of p loop *)
   Print["Log-Log Plot, Base 10, Pull out all powers of 2"];
   Print[
    ListPlot[loglogpowerresults, AxesLabel → {"Log_10(seed)", "Log_10(steps)"}]];
   Print["SemiLog Plot, Base 10, Pull out all powers of 2"];
   Print[ListPlot[semilogpowerresults, AxesLabel → {"Log_10(seed)", "steps"}]];
  ];
```

In[13]:=
```
Timing[iteratefrangeanalyze[100, 10, 200, 2] ]
```

```
We have done 10%.

We have done 20%.

We have done 30%.

We have done 40%.

We have done 50%.

We have done 60%.

We have done 70%.

We have done 80%.

We have done 90%.

We have done 100%.

Log-Log Plot, Base 10, Pull out all powers of 2
```




```
SemiLog Plot, Base 10, Pull out all powers of 2
```




```
Out[13]= {1316.46, Null}
```