We want to know how hard it is to use search to find a preimage for a hash function. We will use the RSA-devised hash function for or experiment here.

**(1)** Choose two very large primes $p$ and $q$, both at approximately size $2^{256}$. Therefore, $n = pq$ should have size about 512 bits. Pick $e$ such that $\gcd(e, phi(n)) = 1$.

- **(a)** Pick a random number $k$ where $1 < k < n$.
- **(b)** Starting with $k$, check whether $k^e$ is between $n/2$ and $n$. If not, try $k + 1$, $k + 2$, until the answer is within the range. Record the number of trials.
- **(c)** Now repeat the experiment to find a value $k$ where $k^e$ is between $n/2$ and $3n/4$.
- **(d)** Similarly, now reduce your range to $n/2$ and $5n/8$.
- **(e)** Repeat and show your findings as the range becomes smaller. Plot a graph.
- **(f)** Can you give a conjecture on the difficulty as the range becomes smaller?

**(2)** Repeat the above experiment. Instead of picking a random $k$ and then increase $k$ as we progress, we pick a random value every time to check the output against the range. Is there a difference?

In general, a hash function should be random enough such that any slight change in the input will create major changes in the output. Once you are done with this module, we will investigate industrially used hash functions and their relative difficulty to find a *pre-image*.