Problem 5: Be Squareful - Solution

There were many possible solution approaches to this problem. The one approach that does not work, is to simply loop over $1 \leq a \leq 2 \cdot 10^6$ and check if a and $a + k$ are squareful. Checking if a number is squareful is difficult, as it basically is a factoring problem. Like factoring problems in general, we can exploit the fact that we are interested in squarefulness over the entire interval $[1, 2 \cdot 10^6]$. All of the solution approaches make use of this in some way, either through sieving out all non-squareful numbers, cheesily precomputing all squareful numbers up to $2 \cdot 10^6$, or constructing the squareful numbers iteratively. We focus on the latter solution.

From any squareful number n, we can construct more squareful numbers by multiplying n with some other number m. For nm to be squareful, all prime factors that do not appear in n should appear at least twice in m. This condition is also sufficient to guarantee squarefulness of nm. We will take this a step further, and for some n and prime p, we generate the squareful number np^2 if $p \nmid n$ and np if $p \mid n$.

More practically, we use a priority queue to loop over these squareful n, as we want to find the *smallest* squareful a such that $a + k$ is squareful as well. Initially, we insert 1, and look at the smallest prime $p = 2$. Next, for any squareful n we get out the queue, we insert either np^2 if $p \nmid n$ or np if $p \mid n$ back into the queue, and also insert n with the next prime p. In this way, we iteratively construct all squareful numbers, starting with their smallest prime factors. As we know the result will always be at most $2 \cdot 10^6$, we do not insert any numbers greater than $2 \cdot 10^6$. This bound also allows us to bound the number of primes to consider, as any prime factor p of a squareful number $n \leq 2 \cdot 10^6$ is at most $\sqrt{2 \cdot 10^6} < 2000$, as $p^2 \mid n$. We can easily compute the primes up to 2000 on the fly, even by naively checking divisors up to $\sqrt{p}$.

To find the answer, when iterating through the priority queue, we keep track of which prior numbers were squarefree and for any n that we retrieve from the priority queue, we check if $n - k$ was squareful as well. If so, we print $n - k$ and return.