1 Main Idea

In this problem we are given a road where we define the western point to be at position 0 and the eastern point to be at the road's length of $L > 0$. There are n houses at positions $x_1, x_2, \ldots, x_n$ where $0 \leq x_1 < x_2 < \ldots < x_n \leq L$. The goal is to place the minimum amount of phone stations such that each house is within k distance of a phone station.

The main idea of the algorithm that solves this efficiently is to find the position of the westernmost house, and then place a phone station as far east as possible while still being within k distance of the westernmost house. Then, move to the next house that is not within k distance of the phone station and repeat the process. The upshot of this algorithm is that a greedy approach is appropriate because the globally optimal solution contains the locally optimal solutions. Specifically, the set of phone stations that cover all houses must contain the phone station that is as far east as possible while still being within k distance of the westernmost house.

2 Pseudocode

Algorithm 1: Phone Station Placement

```
Input: Array of house positions $x_1, x_2, \ldots, x_n$; road length $L$; distance $k$
Output: Array of phone station positions $p_1, p_2, \ldots, p_m$
$P = \emptyset$ ;                                // Set of phone station positions
$i = 1$;
while $i \leq n$ do
    $p_i = \min(x_i + k, L)$;
    $P = P \cup \{p_i\}$;
    while $i \leq n$ and $x_i \leq p_i + k$ do
        $i = i + 1$ ;                           // Find the next house
    end
end
return $P$;
```

3 Proof of Correctness

4 Runtime Analysis

Initializing the set of phone stations P and index i takes $O(1)$ time. The outer loop runs at most n times. Calculating the position of the phone station p_i and adding it to the set P takes $O(1)$ time. The inner loop runs at most n times acroos all iterations of the outer loop, since they share the same index i, meaning that each house is visited at most once. Therefore, this algorithm runs in $O(n)$ time.