Problem Statement

Let us consider a long, straight country road with n houses scattered sparsely along it. (We can picture the road as a long line segment, with a western endpoint and an eastern endpoint, where the western point is at position 0, the eastern point is at position $L > 0$, and the n houses are at positions $x_1, x_2, \ldots, x_n$, respectively, where $0 \le x_1 < x_2 < \cdots < x_n \le L$.) You want to place cell phone base stations at certain points along the road, so that every house is within (that is, less than or equal to) k miles of at least one of the base stations. Design an efficient algorithm that achieves this goal, using as few base stations as possible. Prove the correctness of your algorithm, and analyze its time complexity.

Main Idea

Given that we have n houses at positions $\{x_1, x_2, \ldots, x_n\}$, we start with finding the first house from the west (i.e., from the left) that is not within the range of any base station. We then place a base station at $x_i + k$, where x_i is the position of the house, and k is the range of the base station. After placing the base station, we remove all houses that are within the range k of this base station. We repeat this process until all houses are covered.

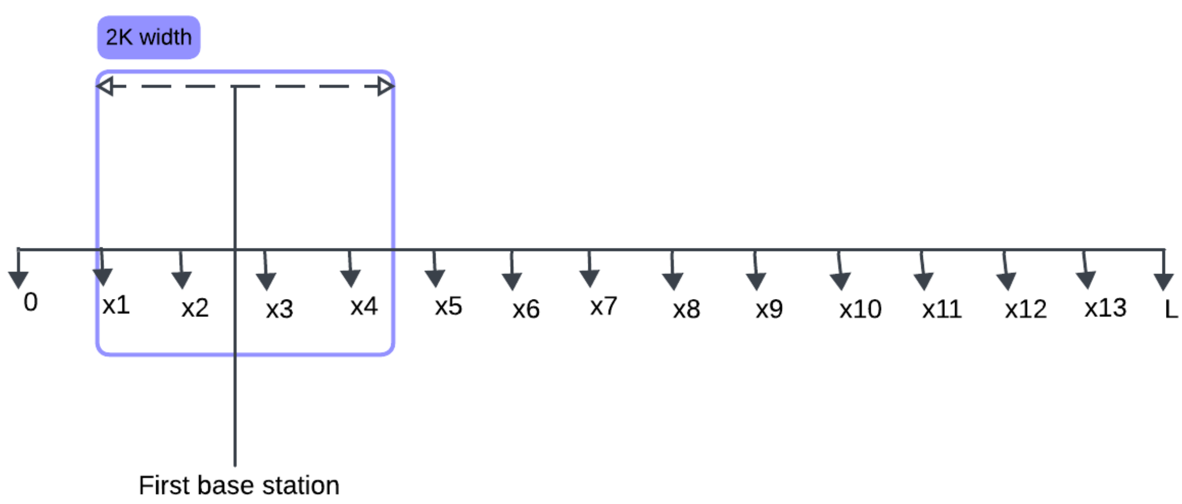

Figure 1: First base station placement

So, for example, in Fig. 1, we first place the base station at a distance K from the first uncovered house, i.e., the first house. It is possible that this base station might cover houses on the right as well. Hence, our next west-most uncovered house is x_5.

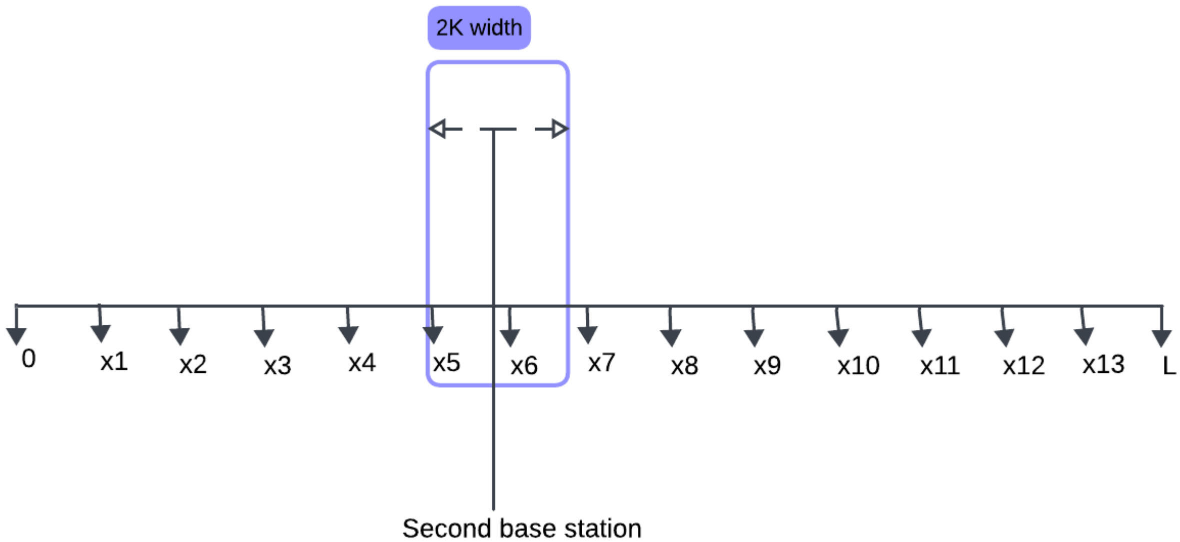

Figure 2: First base station placement

Now, we cover the house at x_5, as it is the westernmost uncovered house by placing the base station at x_5 + k position , and then we proceed in the same way. The next house would be x_7. We keep doing this until we've covered all the houses by placing base stations appropriately. This method will place the minimum number of base stations to cover all houses.

Pseudocode

Algorithm 1 Base Station Placement Algorithm

```
i ← 1
last_base_st_pos ← −∞
Base_st_pos ← {}
while i ≤ n do
    if house_pos[i] ≤ last_base_st_pos + k and house_pos[i] ≥ last_base_st_pos − k then
        i ← i + 1
    else
        last_base_st_pos ← house_pos[i] + k
        Base_st_pos.insert(last_base_st_pos)
    end if
end while
return Base_st_pos
```

Variable	Description
i	Index for the current house being evaluated
$last\_base\_st\_pos$	Position of the last placed base station
$Base\_st\_pos$	Set of positions where base stations are placed
$house\_pos[i]$	Position of the i^{th} house
k	The range of each base station (maximum distance it can cover)
n	Total number of houses along the road

Table 1: Explanation of Variables in the Base Station Placement Algorithm

Proof of Correctness

We will use **proof by contradiction** Let $b_1, b_2, \ldots, b_{m_{\text{greedy}}}$ be the base stations placed by our greedy algorithm, and let $c_1, c_2, \ldots, c_{m_{\text{optimal}}}$ be the base stations placed by the optimal algorithm. We have:

$$b_1 < b_2 < \cdots < b_{m_{\text{greedy}}}$$

$$c_1 < c_2 < \cdots < c_{m_{\text{optimal}}}$$

We assume optimal solution covers all houses and $m_{\text{optimal}} < m_{\text{greedy}}$. We will prove that this assumption leads to a contradiction.

If $m_{\text{optimal}} < m_{\text{greedy}}$, then it means at least some of the base stations placed by the optimal algorithm must cover more houses than their greedy counterparts.

The first base station position placed by the greedy algorithm is $b_1 = x_1 + k$. Now, we say that either $c_1 < b_1$ or $c_1 > b_1$.

- For $c_1 < b_1$:
  c_1 cannot cover more houses than b_1 because on the left side, only one house is uncovered, and on the right side, $c_1 + k < b_1 + k$ (since $c_1 < b_1$), so it covers fewer or same number of houses.

- For $c_1 > b_1$:
  c_1 is not able to cover the first house since $c_1 - k > x_1$ (as $c_1 > b_1$ and $b_1 = x_1 + k$). Since other c_i's are greater than c_1, the first house will remain uncovered, meaning the optimal solution won't be able to cover all houses which renders our assumption of optimal solution covering all houses incorrect.

Therefore, c_1 cannot cover more houses than b_1.

We can extend this argument for any i. Let x_j be the westernmost house not yet covered by any base station. Then:

$$b_i = x_j + k$$

Now, either $c_i < b_i$ or $c_i > b_i$:

- For $c_i < b_i$:
  The range covered by c_i is $[c_i - k, c_i + k]$, which is less than the range covered by b_i, because $c_i < b_i$. Since x_j is the leftmost uncovered house, shifting c_i to the left will not cover more houses (since houses to the left of x_j are already covered), and it will lose coverage on the right side. Hence, the optimal placements cannot cover more houses than the greedy algorithm.

- For $c_i > b_i$:
  The range covered by c_i will not cover x_j, since $c_i - k > x_j$ (as $c_i > b_i$ and $b_i = x_j + k$). This is similar to the first base station placement argument and leads to the assumption of optimal solution covering all houses incorrect.

Thus, c_i does not cover more houses than b_i.

Therefore, this statement holds for all base station placements $i = 1, 2, \ldots, m_{optimal}$, The assumption that the optimal algorithm covers all houses and $m_{\text{optimal}} < m_{\text{greedy}}$ must be false. Hence, the greedy algorithm uses the minimum number of base stations required to cover all houses. Therefore, $m_{\text{optimal}} = m_{\text{greedy}}$.

Time Complexity

In the algorithm, the while loop will at-max run for 2xn times which might be for example each base station covering only one house. Inside the while O(1) complexity calculations are being performed. Therefore, the total complexity will be O(n)