2.3 Art Gallery

Andre Monteiro

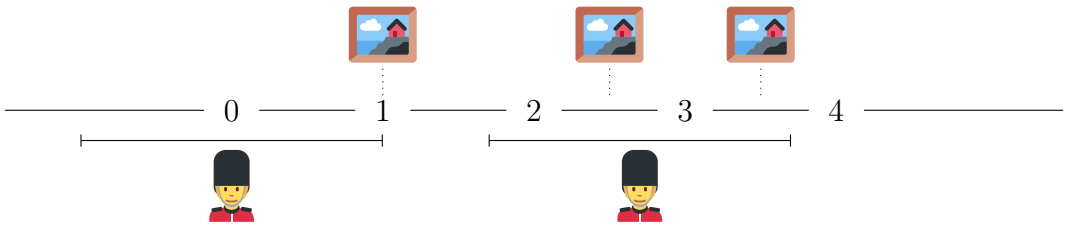
Term 3, 2025

Problem Summary

Given a sorted array

$$P = [p_1, p_2, \ldots, p_n]$$

of n real numbers representing the positions of paintings along a hallway, determine the minimum number of guards required to protect all paintings. Each guard can protect any painting located within a distance of 1 unit to their left or right (i.e., a total coverage of 2 units). The goal is to compute the optimal placement of guards such that all paintings are protected while minimizing the number of guards used.



Question a)

The art gallery also presents Larry with one potential strategy to protect the paintings:

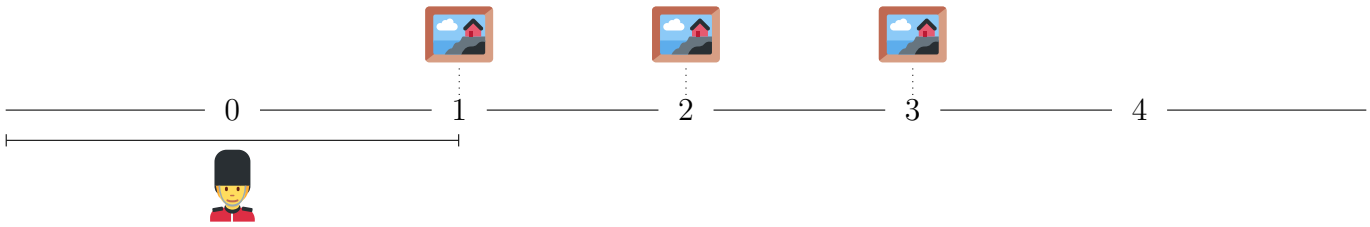
> *"Place a guard one unit to the left of the leftmost unprotected painting, and repeat."*

To demonstrate that the given algorithm does not always yield an optimal solution, consider the following example:

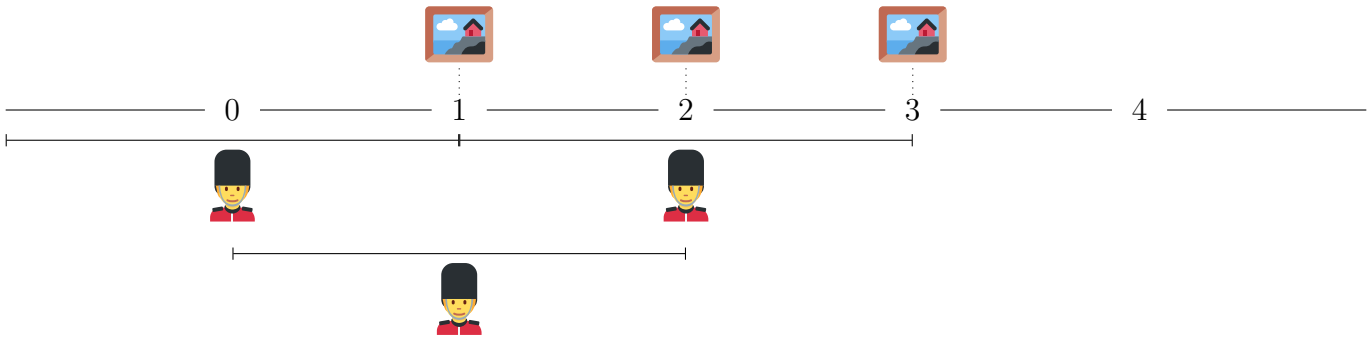
Let $N = 3$, representing 3 paintings, with positions given by the array:

$$P = [1, 2, 3]$$

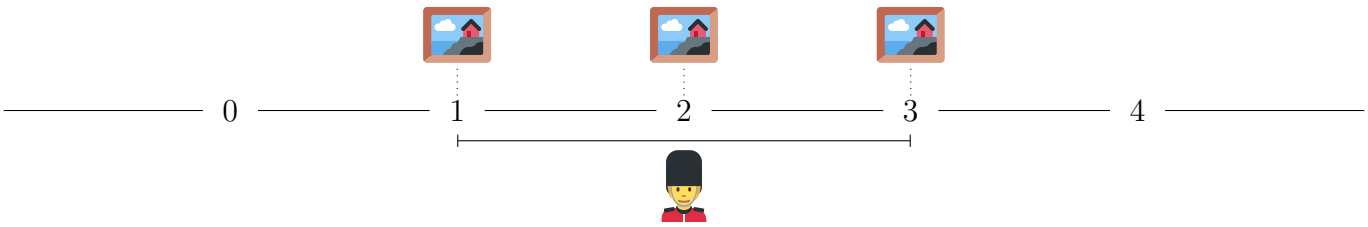
According to the algorithm, we place the first guard *one unit to the left* of the leftmost unguarded painting. Since the first painting is located at position 1, the guard is placed at position 0:



After placing the first guard, the algorithm proceeds by discarding the paintings already covered and repeating the same step for the next leftmost unguarded painting. The process continues as follows:



As a result, the algorithm places **three guards**, one for each painting, even though it is possible to protect all three paintings using only one guard, for example by placing a guard at position 2. This demonstrates that the algorithm is not optimal in all cases.



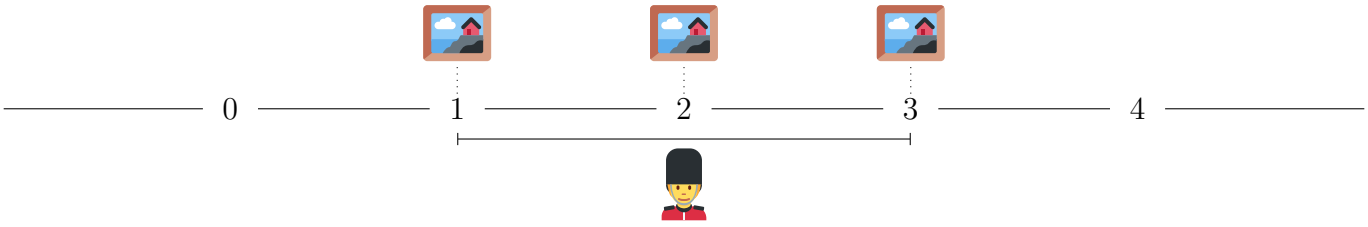
Question b)

To better optimize the provided implementation, the gallery shoudl instead position the guard one unit to the right, as follows:

Let $N = 3$, representing 3 paintings, with positions given by the array:

$$P = [1, 2, 3]$$

After placing the first guard, the algorithm proceeds by removing the paintings already covered and repeating the same process for the next leftmost unguarded painting. This continues until all paintings are protected.



As such, the algorithm places **one guard**, which is indeed the optimal solution.

Correctness Argument (Through induction)

We aim to prove that the greedy algorithm correctly places the minimum number of guards required to protect all paintings, given that each guard covers a range of 2 units (from $x - 1$ to $x + 1$).

Claim: The greedy algorithm always produces an optimal (i.e., minimal) number of guards to cover all paintings in the array $P = [p_1, p_2, \ldots, p_n]$, where p_i represents the position of the i^{th} painting.

Base Case (1 painting): If there is only one painting, the algorithm places one guard at position $p_1 - 1$, which fully covers the painting at p_1. Since at least one guard is needed to cover a painting, this is clearly optimal.

Inductive Hypothesis: Assume that for any list of k paintings (where $1 \leq k < n$), the greedy algorithm places the minimum number of guards necessary to fully cover all k paintings.

Inductive Step: We must show that the algorithm also produces an optimal result for $k + 1$ paintings.

The algorithm examines the leftmost unprotected painting p_i and places a guard at position $p_i + 1$, which covers all paintings in the range $[p_i, p_i + 2]$. It then skips all paintings in that range and repeats the process for the next unprotected painting.

Suppose there exists an alternative solution that uses fewer guards. Then, it must have covered p_i without placing a guard covering $[p_i, p_i + 2]$. But since p_i is the leftmost unprotected painting, and no guard before this step covers it, any solution must place a guard whose range includes p_i. Placing it any farther to the right would miss p_i, and placing it farther to the left would not cover more paintings (due to the sorted nature of the array). Therefore, the greedy placement is locally optimal and does not prevent future optimal placements.

By the inductive hypothesis, the remaining k paintings (after skipping those covered) are also covered optimally.

Conclusion: By induction, the greedy algorithm correctly computes the minimal number of guards required to protect any number of paintings, and thus is optimal.

Time Complexity

The algorithm performs a single pass over the sorted array of painting positions:

- At each step, it selects the leftmost unprotected painting.
- It places a guard to cover as many paintings as possible within the 2-unit range.
- It skips over all paintings that are now protected.

Each painting is checked at most once, and each guard placement is a constant-time operation. There are no nested loops or recursive steps.

Therefore, the total time complexity is:

$$\boxed{O(n)}$$

where n is the number of paintings. Even in the worst-case scenario, where each painting is more than 2 units apart and requires its own guard, the algorithm still only makes one linear pass.