Sample solution

Assume for now that number of nails is a power of 2

1. Divide the nails into two equal parts
2. Weigh each part. One of them will be heavier while the other is gonna be lighter. Use heavier part to calculate correct weight of a single nail. Now the light nail lies in the other part and we would focus on this part.
3. Divide the current solution set into half and weigh. If the weight is equal to the correct weight as computed using the average weight of nail then the solution set is reduced to the other part. Otherwise, the solution set is reduced to the chosen part. In any case, we have halved the solution set.
4. Repeat 3 until you are left with 1 nail in solution set.

Time complexity - $\log_2 n + 1$ (Extra one since initially we did 2 weighings instead of one)

If we relax the assumption that number of nails is power of 2, then time complexity becomes $\text{ceil}(\log_2 n) + 1$

Bonus
We can do slightly better at the top step. Instead of dividing into two equal parts divide into three equal parts. Weigh two parts, if both weigh the same then our solution set is the remaining part otherwise our solution set is the lighter part. From the heavier part we can get the correct weight of nail. Note here in two weighings we divided the set by 3 instead of 2 earlier. From now on repeat step 3 of the above algorithm.

Justification
Argue that we cannot do better than reducing the solution by half at any moment.
We cannot do better since we can imagine the search as a binary tree where each nail is a leaf. At each step we can reduce the search space by half/ The height of the tree is $\log_2 n$ and at each step we can reduce height of search tree by 1. So we cannot do better than $\log_2 n$.

Suppose we do an unequal split instead choosing fraction p. Now our solution set in worst case is max(p,1-p) which is lower bounded by half thus we must do equal splits only.

What if we do k-way split instead of 2-way splits?
Algorithm scales as $\textbf{ceil}(\log_2 k)\ \log_k n$ which is again same as $\log_2 n$ (assuming k is power of 2). Thus 2-way split is as good as we can get.