1884B - Haunted House

In order for a number to be divisible by 2^i, the last i bits of its binary representation must be equal to 0. For convenience, let's reverse the binary representation of the number so that our operations aim to zero out the first i bits. Let $zero$ be the number of bits equal to 0 in the original string.

If $i > zero$, it is obvious that the answer is -1 since our operations do not change the number of zeros and ones in the number. Otherwise, the answer exists, and we will learn how to calculate the minimum number of operations. Consider all $j \le i$, where $s_j = 1$. We need to remove these ones from our prefix by replacing them with the nearest zeros after position i. We will traverse the string from left to right, keeping track of the number of ones in our prefix (denoted as cnt) and the sum of their positions (denoted as sum_one). We also need to maintain the sum of the nearest cnt positions of zeros after our element i (denoted as sum_zero). This can be done using a pointer. The answer is $sum\_zero - sum\_one$. This is both a lower bound estimate and can be greedily shown how to obtain this answer in such a number of operations.

First, let's place all the ones at the end of our prefix. We will place the rightmost one at position i and so on. This can be done in $\sum_{j=i-cnt+1}^{i} (j - pos)$ operations, where pos is the position of the corresponding one, which in general is actually $= -sum\_one + \sum_{j=i-cnt+1}^{i} j$. Now we want to place zeros at these positions, and we will do this greedily cnt times, performing $\sum_{j=i-cnt+1}^{i} (pos - j)$ operations, where pos is the position of 0. Again, the total is $= sum\_zero - \sum_{j=i-cnt+1}^{i} j$. Our answer is the sum of these two, which is $sum\_zero - sum\_one$.