

Problem Title Description. Problem Name: Dyslexic Gollum Author's Name: Palindromic Strings Problem Code: PSTRINGS Alphabet: E

Problem: Given N and K , find how many binary strings of length N have no palindromic string of length $\geq K$.

Solution: The constraints $K \leq 10$ suggests that the solution is pretty lax in terms of bounds on K . If a string has a substring palindrome of length x , then by removing the first and last letter, we get that the string also has a substring of length $x - 2$. This means, if we were to just ensure that our supposed string does not have a palindrome of length K or $K + 1$, then we would be sure that it didn't have a palindrome of any length $\geq K$.

So lets solve this problem by dynamic programming. Ask yourself the question, "Ok, I'm filling in bits from left to right. At the current position I have a choice to either put a 0 or a 1... What information do I need to ensure that I *avoid* all palindromes of length K or $K + 1$?"

Well, you would need to know all the previous K bits of the string. Indeed, if you knew all the previous K bits, then you could decide if you needed to avoid one of 0 or 1, or whether you could carry on.

Which begs the question, "If I knew that the previous K bits were stored in the bitmask bm , and I am at the current position i , how many good strings of length N are possible by filling in bits from i to the end?" Answering this question, would give you your answer.

Let $f(i, bm)$ be as described above. $f(i, bm) = f(i + 1, bm_0)$ [if bm with 0 appended to the end does not cause a palindrome] + $f(i + 1, bm_1)$ [if bm with 1 appended to the end does not cause a palindrome]. Here $bm_0 = (bm \ll 1) \& ((1 \ll K) - 1)$, and $bm_1 = ((bm \ll 1) + 1) \& ((1 \ll K) - 1)$. The base cases are $f(N, bm) = 1$ for all valid bm , and the answer required is $\sum_{bm} f(K, bm)$ over all non-palindromic K -bit bm .

Time Complexity: There are $O(N * 2^K)$ states. It takes $O(K)$ time to check for palindromes. Hence, solution has $O(NK * 2^K)$ time complexity per test-case.