

1400. Construct K Palindrome Strings

Given a string *s* and an integer *k*, return *true* if you can use all the characters in *s* to construct *k* palindrome strings or *false* otherwise.

Example 1:

Input: *s* = "annabelle", *k* = 2

Output: true

Explanation: You can construct two palindromes using all characters in *s*.
Some possible constructions "anna" + "elble", "anbna" + "elle", "anellena" + "b"

Example 2:

Input: *s* = "leetcode", *k* = 3

Output: false

Explanation: It is impossible to construct 3 palindromes using all the characters of *s*.

Example 3:

Input: *s* = "true", *k* = 4

Output: true

Explanation: The only possible solution is to put each character in a separate string.

Constraints:

- $1 \leq s.length \leq 10^5$
- *s* consists of lowercase English letters.
- $1 \leq k \leq 10^5$

1400. Construct K Palindrome Strings

```
/*  
    Math: counting number of odds frequencies  
    using an array for frequencies  
    Time Complexity:  $O(26+n+26)=O(n)$   
    Space complexity:  $O(26)=O(1)$   
*/  
class Solution{  
    public:  
        bool canConstruct(std::string s, int k){  
            if(k>s.size()) return false;  
  
            std::vector<int> freq(26,0);  
            for(auto& letter: s) freq[letter-'a']++;  
  
            int count_odd=0;  
            for(auto& f: freq) count_odd+=(f!=0&&f%2);  
  
            return count_odd<=k;  
        }  
};
```

1400. Construct K Palindrome Strings

```
/*  
    Math: counting number of odds frequencies  
    using a bitmask  
    Time Complexity:  $O(n)$   
    Space complexity:  $O(1)$   
*/  
class Solution {  
    public:  
        bool canConstruct(std::string s, int k) {  
            if(k>s.size()) return false;  
  
            // Update the bitmask for each character  
            int bitmask=0;  
            for(auto& letter: s) bitmask^=1<<(26-(letter-'a'));  
  
            // Count the number of set bits in the bitmask  
            int count_odd= __builtin_popcount(bitmask);  
  
            return count_odd<=k;  
        }  
};
```