

Analysis: Recycled Numbers

Many contestants got stuck in this problem because of the sample test case number 4. Let's say n is 1212, then after moving 1 or 3 digits you will get 2121, hence the pair (1212, 2121) will be counted twice if you count all possible moves. You can avoid this by breaking out of the loop once you reach the original number again, which will happen after moving 2 digits in the above example.

For the small input you can simply check for each pair (n, m) , $A \leq n < m \leq B$, whether it satisfies the conditions mentioned in the problem statement, and whether you can obtain m by moving some digits from the back of n to the front without changing their order. To check if you can obtain m from n simply try to move all possible groups of digits from n and check if what you get is m . The digit shifting can be done using string manipulation or mathematical expressions, both will run within the time limit.

But the previous solution will not run within the time limit for the large test cases. So here is another solution which should run within the time limit. For each number n , $A \leq n \leq B$, try to move all possible groups of digits from its back to its front and check if the resulting number satisfies the conditions or not. If it does satisfy the conditions then increment the result. Don't forget to avoid counting the same number twice.

Here is a sample solution:

```
int solve(int A, int B) {
    int power = 1, temp = A;
    while (temp >= 10) {
        power *= 10;
        temp /= 10;
    }
    int result = 0;
    for (int n = A; n <= B; ++n) {
        temp = n;
        while (true) {
            temp = (temp / 10) + ((temp % 10) * power);
            if (temp == n)
                break;
            if (temp > n && temp <= B)
                result++;
        }
    }
    return result;
}
```