COUNT NUMBERS SMALLER THAN 10^9 WITH A GIVEN DIGIT SUM

COURSE WORK #11

Author: prof. Yevhenii Borodavka

PROBLEM STATEMENT

You have an integer number **S** (1<=**S**<=81). You need to count numbers smaller than 1 000 000 000 with digit sum **S**.

Input. The single number **S**.

Output. The count numbers smaller than 1 000 000 000 with digit sum **S**.

Example

Input: 1

Output: 10

THE PROBLEM SOLVING

We can do a digit DP having state (current index, whether the currently constructed number of i digits is equal or less than the number formed by the first i digits of N, the sum of digits of the currently constructed number). Let dp[i][tight][sum_so_far] denote the count of numbers whose first i digits have been considered and tight denotes whether the currently constructed number is equal to or less than the number formed by the first i digits of N. If tight is false, then it means that the currently constructed number equals the number constructed by the first i digits of N. If it is true then it means that the currently constructed number is less than the number constructed by the first i digits of N. sum_so_far denotes the sum of digits of the currently constructed number.

THE PROBLEM SOLVING

Base Case: If i = number of digits in N, sum_so_far = Sum, then the answer is 1 else answer is 0.

Transitions: For filling (i+1)th digit we can consider the following — If tight is false, then it means that our constructed number is still equal to the number formed by the first i digits of N. We can try our current possible digit value from 0 to (i+1)th digit of N. If we try the digit more than (i+1)th digit, then the constructed number will become greater than the number formed by the first i digits of N, which will violate the property that our constructed number should be <= N.

THE PROBLEM SOLVING

If tight is true, then it means that the number constructed from the first (i-1) digits has become less than the number constructed from the first (i-1) digit of **N**. So it means that our number can never exceed **N**, so we can choose any digit from 0 to 9. **Nth** sum_so_far can be obtained by adding sum_so_far and the current digit (curr_digit). Finally, we will return the answer which is the count of numbers up to **N** that have a digit sum equal to **S**.

THE PROBLEM SOLVING

Let's consider a simple example with **N**=24 and **S**=4. We need to count all numbers from 0 to 24 whose digit sum is equal to 4. For this simple example, we know the correct answer is **3** (4, 13, and 22). Now let's consider the algorithm step by step.

Step 0. Input data, DP tables for tight=false and tight=true.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	-1	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	-1	-1	-1	-1

THE PROBLEM SOLVING

Step 1. Current id=0, tight=false and sum_so_far=0. We try the digit 0 as the first digit in the constructed number. We fix this digit and try to change the second digit. Tight for the next digit will be true because the first digit zero is less than the first digit in the number 2. The sum_so_far for the second digit is the current sum plus the current digit (0+0=0).

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	0	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	-1	-1	-1	-1

THE PROBLEM SOLVING

Step 2. Current id=1, tight=true and sum_so_far=0. We try the digit 0 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+0) is equal to **S** or not. For the current digit 0, it is not equal to 4. So, next, we try the digit 1 as the second digit of the number.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	0	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	0	-1	-1	-1

THE PROBLEM SOLVING

Step 3. Current id=1, tight=true and sum_so_far=0. We try the digit 1 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+1) is equal to **S** or not. For the current digit 1, it is not equal to 4. So, next, we try the digit 2 as the second digit of the number.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	0	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	0	-1	-1	-1

THE PROBLEM SOLVING

Step 4. Current id=1, tight=true and sum_so_far=0. We try the digit 2 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+2) is equal to **S** or not. For the current digit 2, it is not equal to 4. So, next, we try the digit 3 as the second digit of the number.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	0	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	0	-1	-1	-1

THE PROBLEM SOLVING

Step 5. Current id=1, tight=true and sum_so_far=0. We try the digit 3 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+3) is equal to **S** or not. For the current digit 3, it is not equal to 4. So, next, we try the digit 4 as the second digit of the number.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	0	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	0	-1	-1	-1

THE PROBLEM SOLVING

Step 6. Current id=1, tight=true and sum_so_far=0. We try the digit 4 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+4) is equal to **S** or not. For the current digit 4, it is equal to 4. So, we update the value in the table for id=1, tight=true, and sum_so_far=0.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	0	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	-1	-1	-1

THE PROBLEM SOLVING

Step 7. Current id=1, tight=true and sum_so_far=0. As tight=true we continue trying the digit up to 9 as the second digit in the constructed number. No other digit matches the S value, so we finish with the second digit and go back up to the first digit.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	0	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	-1	-1	-1

THE PROBLEM SOLVING

Step 8. Current id=0, tight=false and sum_so_far=0. After we try 0 as the first digit of the constructed number, we get the result 1 — which means two-digit numbers starting with zero have only one number whose digit sum is 4. Now we try 1 as the first digit. We fix this digit and try to change the second digit. Tight for the next digit will be true, and sum_so_far (0+1=1).

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	1	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	-1	-1	-1

THE PROBLEM SOLVING

Step 9. Current id=1, tight=true and sum_so_far=1. We try the digit 0 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+0) is equal to **S** or not. For the current digit 0, it is not equal to 4. So, next, we try the digit 1 as the second digit of the number.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	1	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	0	-1	-1

THE PROBLEM SOLVING

Step 10. Current id=1, tight=true and sum_so_far=1. We try the digit 1 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+1) is equal to **S** or not. For the current digit 1, it is not equal to 4. So, next, we try the digit 2 as the second digit of the number.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	1	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	0	-1	-1

THE PROBLEM SOLVING

Step 11. Current id=1, tight=true and sum_so_far=1. We try the digit 2 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+2) is equal to **S** or not. For the current digit 2, it is not equal to 4. So, next, we try the digit 3 as the second digit of the number.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	1	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	0	-1	-1

THE PROBLEM SOLVING

Step 12. Current id=1, tight=true and sum_so_far=1. We try the digit 3 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+3) is equal to **S** or not. For the current digit 3, it is equal to 4. So, we update the value in the table for id=1, tight=true, and sum_so_far=1.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	1	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	1	-1	-1

THE PROBLEM SOLVING

Step 13. Current id=1, tight=true and sum_so_far=1. As tight=true we continue trying the digit up to 9 as the second digit in the constructed number. No other digit matches the S value, so we finish with the second digit and go back up to the first digit.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	1	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	1	-1	-1

THE PROBLEM SOLVING

Step 14. Current id=0, tight=false and sum_so_far=0. After we try 1 as the first digit of the constructed number, we get the result 1 — which means two-digit numbers starting with one have only one number whose digit sum is 4. Now we try 2 as the first digit. We fix this digit and try to change the second digit. Tight for the next digit will be false, because 2 is equivalent to the first digit in the number, and sum_so_far (0+2=2).

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	2	-1	-1	-1
1	-1	-1	-1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	1	-1	-1

THE PROBLEM SOLVING

Step 15. Current id=1, tight=false and sum_so_far=2. We try the digit 0 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+0) is equal to **S** or not. For the current digit 0, it is not equal to 4. So, next, we try the digit 1 as the second digit of the number.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	2	-1	-1	-1
1	-1	-1	0	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	1	-1	-1

THE PROBLEM SOLVING

Step 16. Current id=1, tight=false and sum_so_far=2. We try the digit 1 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+1) is equal to **S** or not. For the current digit 1, it is not equal to 4. So, next, we try the digit 2 as the second digit of the number.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	2	-1	-1	-1
1	-1	-1	0	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	1	-1	-1

THE PROBLEM SOLVING

Step 17. Current id=1, tight=false and sum_so_far=2. We try the digit 2 as the second digit in the constructed number. As the second digit is the last one in the number we check if (sum_so_far+2) is equal to **S** or not. For the current digit 2, it is equal to 4. So, we update the value in the table for id=1, tight=false, and sum_so_far=2.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	2	-1	-1	-1
1	-1	-1	1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	1	-1	-1

THE PROBLEM SOLVING

Step 18. Current id=1, tight=false and sum_so_far=2. As tight=false we continue trying the digit up to 4 as the second digit in the constructed number. No other digit matches the S value, so we finish with the second digit and go back up to the first digit.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	2	-1	-1	-1
1	-1	-1	1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	1	-1	-1

THE PROBLEM SOLVING

Step 19. Current id=0, tight=false and sum_so_far=0. After we try 2 as the first digit of the constructed number, we get the result 1 — which means two-digit numbers starting with two have only one number whose digit sum is 4. As tight=false and the first digit in the number is 2, we can not try any other digits, so we finish and return the current result 3.

Input data		
Indexes	0	1
Digits	2	4

tight=false				
id	0	1	2	3
0	3	-1	-1	-1
1	-1	-1	1	-1

tight=true				
id	0	1	2	3
0	-1	-1	-1	-1
1	1	1	-1	-1

THANK YOU!