1 Miscellaneous Topics

1.1 IO

LeetCode implicitly uses cin/cout IO when testing your code. To speedup the IO, you can use the following code globally:

```
//IO
int _IO=[](){
    std::ios::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    return 0;
}();
```

1.2 running time

The final running time is the total running time over all test cases.

1.3 global variables

be careful to initialize them.

1.4 multithreading

LeetCode supports multithreading. For example, see the 292ms code for 318. Maximum Product of Word Lengths. However, there are many (small) test cases, so multithreading usually makes the code slower.

1.5 template meta-programming

LeetCode supports template meta-programming. However, if the compile time is too long, you will get Compile Error: Compile time limit exceeded. see the 60ms code for 338. Counting Bits.

References