1 Miscellaneous Topics

Let's forget about the theoretical solutions for a while and get hands dirty. In the following, the default language we use is C++.

1.1 IO

LeetCode implicitly uses cin for 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);
    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 manually.

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.

1.6 assembly language

LeetCode supports inline assembly language. see the 20ms code for 307. Range Sum Query - Mutable.

1.7 hack the Online Judge system

When you submit your code, LeetCode enables you to use many powerful functions, so we can do interesting things. For example, we can replace the LeetCode cin IO by our hand-written faster IO, and significantly improve the total running time. see the 4ms code for 307. Range Sum Query - Mutable.
Use the following code to use your own main function and overide the (hidden) main function provided by the judge system:

```
//main
int _main=[](){
    FILE *fout=fopen("./user.out","w");
    exit(0);
    return 0;
}();
```

useful tools in C++:

```
#include<bin/bash>
#define Dbg(x) cout<<"debug: "<<__FUNCTION__<<"() @ "<<__TIMESTAMP__<<"\n"\
        <<__FILE__<<" L"<<__LINE__<<"\n"<<#x" = "<<(x)<<endl
getcwd(buffer,255)
system("cd / && ls -al")
pthread\_create(&p[i],NULL,func,(void*)&(ids[i]))
FILE *fout=fopen("./user.stdout","w")
__asm__("test %%eax,%%eax\n")
```

for more information, see template.cpp.

References