

FZU 1759 Super A^B mod C (欧拉函数,快速幂,降幂公式) - CSDN博客

题目链接: <http://acm.fzu.edu.cn/problem.php?pid=1759>

一道吓人的题。。

不禁再次感叹数学真伟大,使用下面的降幂公式很简单就写出来了。

$$A^B \bmod C = A^{B \bmod \varphi(C) + \varphi(C)} \bmod C$$

phi是欧拉函数,如果不太清楚欧拉函数是什么,怎么求欧拉函数,可以看看下面这两个博客,或者参考维基百科。

http://blog.csdn.net/leolin_/article/details/6642096

http://blog.csdn.net/once_hnu/article/details/6302868 (直接法和筛选法求欧拉函数值)

学会了求欧拉函数值,我们就可以利用上面那个降幂公式来计算结果了。

参考博客: <http://blog.csdn.net/t1019256391/article/details/37595993> 感谢!

[cpp]view plaincopy

```
1. #include
2. #include
3. #include
4. #include
5. using namespace std;
6. typedef long long ll;
7. ll phi(ll n) { //直接法求欧拉函数值
8. int res = n, a = n;
9. int i;
10. for(i = 2; i * i <= a; i++) {
11. if(a % i == 0) {
12. res -= res / i;
13. while(a % i == 0) a /= i;
14. }
15. }
16. if(a > 1) res -= res / a;
17. return res;
18. }
19. ll qpow(ll a, ll b, ll c) { //快速幂
20. ll res = 1;
21. while(b) {
22. if(b & 1) res = res * a % c;
23. a = a * a % c;
24. b >>= 1;
25. }
26. return res;
27. }
28. int main() {
29. ll a, c;
30. char b[1000010];
31. while(~scanf("%I64d %s %I64d", &a, b, &c)) {
32. ll phic = phi(c);
33. a %= c;
34. int i, len = strlen(b);
35. ll res = 0;
36. for(i = 0; i < len; i++) {
37. res = res * 10 + b[i] - '0';
38. if(res > phic) break; //降幂公式的条件,只有指数大于phi(c)才可用
39. }
40. if(i == len) {
41. printf("%I64d\n", qpow(a, res, c)); //指数小于等于phi(c),直接计算
42. }
43. else {
44. res = 0; //降幂
45. for(i = 0; i < len; i++) {
```

```
46.         res = res * 10 + b[i] - '0';
47.         res %= phic;
48.     }
49.     printf("%I64d\n", qpow(a, res + phic, c));
50. }
51. }
52. return 0;
53. }
```