2023/05/11

一：Leftmost Digit

分析

题目就是求出n^n最左边的数字是多少。

考虑到数字非常大，并且我们只需要求最左边的第一位，即第一眼就可以想到科学计数法。

即

- $n^n = a \cdot 10^m$
- 取对数得：$nlgn = m + lga$
- 我们要求的应该是a的整数部分。
- 求出a即可。
- $lga = nlgn - m$
- 现在就剩下m不知到是谁了，如果能够把m转换成和n有关的式子题目就做完了。
- m表示科学计数法的位数，我们知道$lg10 = 1$
- $1 \le a \le 9$所以$0 < lga < 1$
- 所以设$a = 10^x$其中$0 < x < 1$
- 所以$n^n = 10^x \cdot 10^m = 10^{x+m}$
- 所以$nlgn = x + m$
- 即$m = \lfloor nlgn \rfloor$

`AC Code`

```
#include <cmath>
#include <iostream>
#include <cstdio>

using namespace std;

int main(){
    int T;
    scanf("%d", &T);
    while(T -- ){
        int n;
        scanf("%d", &n);
        double x = n * log10(n);
        x -= (long long)x;
        printf("%d\n", (int)pow(10.0, x));
    }
    return 0;
}
```

二：圆桌会议

分析：

首先不考虑环，考虑一个序列，由于只能交换相邻，所以需要$\frac{n\cdot(n-1)}{2}$次

例如：1 2 3 4 最后会变成 4 3 2 1

- 1 2 3 4
- 2 `1` 3 4
- 2 3 `1` 4
- 2 3 4 `1`
- 3 `2` 4 1
- 3 4 `2` 1
- 4 `3` 2 1
- 我们可以发现`1`换了`3`次，`2`换了`2`次，`3`换了`1`次，所以换的次数是一个等差数列求和（首项是`1`，末项是`n-1`，公差是`1`，项数是`n-1`）
- $S_n = n\cdot\frac{a_1+a_n}{2} = n\cdot a_1 + \frac{n(n-1)}{2}$

环也如此吗？举个例子

1 2 3 4 最后会变成 1 4 3 2

我们发现我们只是换了`1 2`和`3 4`

1 2 3 4 5最后会变成`1 5 4 3 2`

发现我们只是换了`4 5 1`和`2 3`

所以我们是把原来序列划分成两个尽量相等的部分，每部分内部交换即可。

`AC Code`

```
#include <bits/stdc++.h>

using ll = long long;

int solve(int x){
    return x * (x - 1) / 2;
}

int main(){
    int n;
    int ans;
    while(std::cin >> n){
        ans = solve(n / 2) + solve(n - n / 2);
        std::cout << ans << '\n';
    }
    return 0;
}
```

2023/05/16

一：找新朋友

分析：

题目就是让我们找与n互质的数的个数。

欧拉函数板子

欧拉函数定义：

$1-N$中与N互质的数的个数被称为欧拉函数，记为$\phi(N)$

若在算数基本定理中，$N=p_1^{\alpha_1}\cdot p_2^{\alpha_2}\cdot\ldots\cdot p_m^{\alpha_m}$ 则:

$\phi(N)=N\cdot(1-\frac{1}{p_1})\cdot(1-\frac{1}{p_2})\cdot\ldots\cdot(1-\frac{1}{p_m})=N\cdot\frac{p_1-1}{p_1}\cdot\frac{p_2-1}{p_2}\cdot\ldots\cdot\frac{p_m-1}{p_m}$

用`容斥原理`证明。

`AC Code`

```
#include <bits/stdc++.h>

using namespace std;

int Euler(int n){
    int res = n;
    for(int i = 2; i <= n / i; i ++ ){
        if(n % i == 0){
            res = res / i * (i - 1);
        }
        while(n % i == 0){
            n /= i;
        }
    }
    if(n > 1){
        res = res / n * (n - 1);
    }
    return res;
}

int main(){
    int T;
    cin >> T;
    int n;
    while(T --){
        cin >> n;
        printf("%d\n", Euler(n));
    }
    return 0;
}
```

奇偶位互换

```cpp
#include <bits/stdc++.h>

using namespace std;

int main(){
    int T;
    cin >> T;
    while( T -- ){
        string s;
        cin >> s;
        int n = s.size();
        for(int i = 0; i < n; i += 2 ){
            swap(s[i], s[i + 1]);
        }
        cout << s << '\n';
    }
    return 0;
}
```

素数对

```cpp
#include <bits/stdc++.h>

using namespace std;

int last[10010];

int main(){
    int cnt = 0;
    for(int i = 2; i <= 10000; i ++ ){
        bool ok = false;
        for(int j = 2; j <= i / j; j ++ ){
            if(i % j == 0){
                ok = true;
                break;
            }
        }
        if(!ok){
            last[++ cnt] = i;
        }
    }
    int n;
    while(cin >> n){
        int dist = 100000;
        int x = 0, y = 0;
        for(int i = 1; i <= cnt; i ++ ){
            for(int j = i; j <= cnt; j ++ ){
                //cout << last[i] << " " << last[j] << endl;
                if(last[i] + last[j] == n){
                    if(last[j] - last[i] < dist){
                        dist = last[j] - last[i];
                        x = last[i];
                        y = last[j];
                    }
```

```
                }
            }
        }
        cout << x << " " << y << '\n';
    }
    return 0;
}
```

```
#include <bits/stdc++.h>

using namespace std;

bool last[10010];

int main(){
    int cnt = 0;
    for(int i = 2; i <= 10000; i ++ ){
        bool ok = false;
        for(int j = 2; j <= i / j; j ++ ){
            if(i % j == 0){
                ok = true;
                break;
            }
        }
        if(!ok){
            last[i] = true;
        }
    }
    int n;
    while(cin >> n){
        int dist = 10000, x, y;
        for(int i = 2; i <= n; i ++ ){
            if(last[i] && last[n - i] && abs(n - i - i) < dist && i <= n - i){
                x = i;
                y = n - i;
            }
        }
        cout << x << " " << y << '\n';
    }
    return 0;
}
```