

NTUT_King ICPC Team Notebook

Contents

1 官-數學

1.1	找因數	1
1.2	公因數	1
1.3	求導數	1
1.4	Sum of Product	1
1.5	法里數列	1

2 官-基礎

2.1	C++ 基礎	1
2.2	C++ 易忘的內建函數	1
2.3	python 內建大數、易錯事項	2

1 官-數學

1.1 找因數

```
int num = 某數;
for(int i = 1; i <= (int)sqrt(num); i++)
{
    if(num % i == 0)
    {
        // 得到因數i
        // 4 = 2 * 2, 只保留一個2
        if(num / i != i)
        {
            // 12 = 3 * 4, 像這樣, 只要找到3, 就知道4
            // 得到因數num / i
        }
    }
}
```

1.2 公因數

```
// * GCD(a 1 + b j, ..., a n + b j)
// = GCD(a[0] + b[j], a[1] - a[0], a[2] - a[1], ....., a[n - 2] - a[n - 3], a[n - 1] - a[n - 2])
// , a[0] < a[1] < a[2] < a[3]...
// 用法
// * 因為只有b j 是不固定的, 所以求GCD(a 1 + b j, ..., a n + b j) 就只要算GCD(a[0] + b[j], baseGcd)
// * baseGcd = GCD(a[1] - a[0], a[2] - a[1], ....., a[n - 2] - a[n - 3], a[n - 1] - a[n - 2])

#include <iostream>
#include <map>
#include <queue>
#include <bits/stdc++.h>
using namespace std;
const int maxn = 2e5 + 10;
const int mod=1e9+7;
typedef long long ll;
map<string, int>mp;
int gcd(int a,int b){
    return b==0?a:gcd(b,a%b);
}
ll a[maxn];
int main() {
    int n,m;cin>>n>>m;
    for(int i=0;i<n;i++) cin>>a[i];
    sort(a,a+n);
    ll g=0;
    for(int i=1;i<n;i++) g=__gcd(a[i]-a[i-1],g);
    for(int i=0;i<m;i++){
        ll x;cin>>x;
        cout<<__gcd(x+a[0],g)<<" ";
    }
    return 0;
}
```

1.3 求導數

* 輸入 $f(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$
 * 求導數 $f'(x) = a_0nx^{n-1} + a_1(n-1)x^{n-2} + \dots + a_{n-1}$
 * 將公式重組, 可以省略多餘的次方運算
 * 如此反覆提取公因數, 最後將函數化為 $x f(x) = (((a_{nx} + a_{n-1})x + a_{n-2})x + \dots + a_1)x + a_0$

1.4 Sum of Product

n = 3
 SOP(Pk) = {0, 1, 2} 內部可調換()

P_k Permutation SOP(P_k)
 P1 0 1 2 2
 P2 0 2 1 2
 P3 1 0 2 2
 P4 1 2 0 2
 P5 2 0 1 2
 P6 2 1 0 2輸出有幾種

SOP(Pk)

* 網路上有公式和序列
 * 1, 1, 1, 3, 8, 21, 43, 69, 102, 145, 197, 261, 336, 425, 527, 645, 778, 929, 1097, 1285, 1492, 1721, 1971, 2245, 2542, 2865, 3213, 3589, 3992, 4425, 4887, 5381, 5906, 6465, 7057, 7685, 8348, 9049, 9787, 10565, 11382, 12241, 13141, 14085, 15072, 16105
 * For n >= 7
 * a(n) = (n^3-16*n+27)/6 (n odd)
 * a(n) = (n^3-16*n+30)/6 (n even)

1.5 法里數列

* 求 Farey sequences(Fn) 的第 k 個分數
 * 利用遞推式不斷求出新的數值
 * 遞推式：
 * \$num = \dfrac{n + b1}{b2};\$
 a3 = num * a2 - a1;
 b3 = num * b2 - b1;

2 官-基礎

2.1 C++ 基礎

```
// * define int long long 避免溢位問題
// * cin\cout 在測資過多時最好加速
// * define debug 用來測試

#include <bits/stdc++.h>
#define int long long
#define debug

using namespace std;

main()
{
    #ifdef debug
    freopen("in1.txt", "r", stdin);
    freopen("out1.txt", "w", stdout);
    #endif // debug
    // 讀寫加速
    // 關閉iostream 物件和cstdio 流同步以提高輸入輸出的效率
    ios::sync_with_stdio(false);
    // 可以通過tie(0) (0表示NULL) 來解除cin 與cout 的繫結, 進一步加快執行效率
    cin.tie(0);
}
```

2.2 C++ 易忘的內建函數

```
# 易忘的內建函數
## 輸入輸出
* gets(char*)
* sscanf(char*, "%d:%d:%d %lf", &h, &m, &s, &speed_new)
%d 表示 int
%lf 表示 double
* printf("%.2d:%.2d:%.2d %.2lf km\n",h,m,s,din)
.2 表示保留 2 位小數(%d 是整數,會自動捨去小數)

### 資料型別
* string = to_string(int)
* int = atoi(string.c_str())

### 運算
* lower_bound(begin, end, num)
  * 從陣列的 begin 位置到 end - 1 位置二分查詢第一個大於或等於 num 的數字,找到返回該數字的地址,不存在則返回 end
  * 通過返回的地址減去起始地址,得到找到數字在陣列中的下標 begin
* __builtin_popcount(int)
  * 回傳整數轉成二進值時所包含 1 的數量
* ^
  * 互斥或 xor 運算子
```

2.3 python 內建大數、易錯事項

```
# # Python 常用程式碼
```

```
## Python 內建大數
# * 可以直接用int() 和各個運算子計算
# * 雖然Python 有BigInt(), 但用不到

from sys import stdin, stdout

def main():
    n = int(stdin.readline())
    for i in range(n):
        line = stdin.readline().split("/")
        # 可直接轉換成大數
        p = int(line[0])
        q = int(line[1])

        # 求最大公因數
        gcdNum = gcd(p, q)

        stdout.write(str(gcdNum))
        stdout.write("\n")

main()

## 易錯事項
# * / 除法運算,結果總是返回浮點型別
# * // 取整除,結果返回捨去小數部分的整數
# * stdout.write(str(p)) 不能沒有str()
# * write 只能輸出字串
```