

2020牛客暑期多校训练营（第七场）

Results

Summary

- Solved 4 out of 10 problems
- Rank 56 / 1140 in official records
- Solved 6 out of 10 afterwards

#	Who	=	Penalty	A	B	C	D	E	F	G	H	I	J	Dirt
7	大吉大利，今晚吃 mian(); 4	246			+	01:03	-4	+1			+1		+	33%
							00:07				00:37		01:39	2/6

Member Distribution

Solved	A	B	C	D	E	F	G	H	I	J
Pantw				√				√		√
Withinlover	O		O							
Gary		√							O	

(√ for solved, O for unsolved, - for tried but not solved)

Solutions

A

应当学学打表的新姿势了（

赛后过题 + 1

直接暴搜的复杂度太高了，然后就贪心的想最大值可能都是十分接近边界的点

于是想到按点到原点的距离排个序，只用前30个搜

然后愉快的WA了

后来改成40个过了（据说可以直接搜过去然而没有成功）

B

尽量选最大的吧 具体没太证

```
void work(int n,int m){
    while(n&&m){
        if(n<m) swap(n,m);
        if(n%m==0){
            for(int i=1;i<=n;i++) tmp[++cnt]=m;
            n-=n;
        }
        else{
            for(int i=1;i<=m;i++) tmp[++cnt]=m;
            n-=m;
        }
    }
}
```

C

主要是1和3操作，2操作单独记一下就解决了

1操作可以转化到根节点上，但是会把根节点到被操作节点的整颗子树的答案算错。修正的方法是加一个等差数列，由于只会查询单点的值所以可以差分一下变成区间加法和单点查询。

D

本来想写 py 后来想一想直接猜只有 $n=1$ 和 $n=24$ 这两个解，就直接过了

E

F

G

H

这个题就是求

$$\sum_{k=1}^n \left(2 \left\lfloor \frac{n}{k} \right\rfloor + \left[n \bmod k \neq 0 \right] \right)$$

直接数论分块即可。

dp如代码所示，主要写下g数组怎么求

对于n个节点的无根树求 $\sum d_i^2$

可以枚举数字在prufer序列出现的次数i从n-2个节点中选出i个的方案为 C_{n-2}^i 这i个位置全填1,2...n都是可行的，剩余的n-i-2个位置用剩下的n-1个数随便填满即可，方案数为 $\{(n-1)\}^{n-2-i}$

故 $\sum d_i^2 = C_{n-2}^i \times n \times \{(n-1)\}^{n-2-i} \times \{(i+1)\}^2$

```

for(int n=2;n<N;n++){ //n 无根树 value
    for(int i=0;i<=n-2;i++){
        (g[n]+=C(n-2,i)*n%M*(i+1)%M*(i+1)%M*poww(n-1,n-2-i))%=M;
    }
for(int n=2;n<N;n++){ //n 无根森林 方案
    for(int i=1;i<=n;i++){
        (f[n]+=C(n-1,i-1)*poww(i,i-2)%M*f[n-i]%M)%=M;
    }
for(int n=2;n<N;n++){ //n 无根森林 value
    for(int i=1;i<=n;i++){
        (A[n]+=C(n-1,i-1)*((A[n-i]*poww(i,i-2)%M+f[n-i]*g[i]%M)%M)%M)%=M;
    }
}

```

直接按题意模拟即可。

```

struct Statement {
    enum Type {
        Alloc, Assign, Store, Load
    } type;
    struct Operand {
        enum Type {
            Variable, Object, Field
        } type;
        char str[3];
        int p0, p1;
        void determine() {
            if(str[1] == '.') type = Field, p0 = str[0] - 'A', p1 = str[2] - 'a';
            else if(islower(str[0])) type = Object, p0 = str[0] - 'a';
            else type = Variable, p0 = str[0] - 'A';
        }
    } left, right;
}

```

```
void determine() {
    if(left.type == Operand::Variable && right.type == Operand::Object)
type = Alloc;
    else if(left.type == Operand::Variable && right.type ==
Operand::Variable) type = Assign;
    else if(left.type == Operand::Field && right.type ==
Operand::Variable) type = Store;
    else if(left.type == Operand::Variable && right.type ==
Operand::Field) type = Load;
}
int execute() {
    int ret = 0, pop;
    switch(type) {
        case Alloc:
            if(A[left.p0] & (1 << right.p0)) ret = 0;
            else A[left.p0] |= (1 << right.p0), ret = 1;
            break;
        case Assign:
            ret = __builtin_popcount(A[left.p0]);
            A[left.p0] |= A[right.p0];
            ret = __builtin_popcount(A[left.p0]) - ret;
            break;
        case Store:
            for(int i = 0; i < 26; ++i) {
                if(A[left.p0] & (1 << i)) {
                    pop = __builtin_popcount(G[i][left.p1]);
                    G[i][left.p1] |= A[right.p0];
                    ret += __builtin_popcount(G[i][left.p1]) - pop;
                }
            }
            break;
        case Load:
            pop = __builtin_popcount(A[left.p0]);
            for(int i = 0; i < 26; ++i) {
                if(A[right.p0] & (1 << i)) {
                    A[left.p0] |= G[i][right.p1];
                }
            }
            ret = __builtin_popcount(A[left.p0]) - pop;
            break;
    }
    return ret;
}
} prog[233];
```

Comments

ptw:

- 希望以后自己推式子的时候仔细一点，别再漏项了 (l)
- 早点想 C 或许就过了，以及板子的正确性是很重要的
- A 的暴力为啥最后没打呢

Withinlover:

- A题可以打表，写C之前应该把A的暴力先调出来的（
- 对板子不太熟悉了C调了好久，赛后过题（

Gary:

- l题逆元求错了（真的zz
- 需要好好维护一下板子库

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:mian:nowcoder_training:2020_multi-university_training_contest_7

Last update: 2020/08/04 19:18