A

- 题意:有个人爱爬楼，让你帮他模拟一下。
- 题解:好累啊。

B

- 题意:求一下调和级数。
- 题解:模拟即可。

C

- 题意:在$2 \times n$的方格上进行数次操作，每次放岩浆或删除岩浆，问每次操作后能否从左上角走到右下角。保证起点和终点没有岩浆。
- 题解:如果有任意两个处于不同行的岩浆他们处于相邻和相同的列，那么就无法到达，否则可以到达，维护即可。

D

- 题意:给定点(x_0, y_0)，按照这个递推式$(a_x \cdot x_{i-1} + b_x, a_y \cdot y_{i-1} + b_y)$构造出数个点，问从$(x_s, y_s)$开始经过$t$的距离最多能经过几个上述点，$(1 \leq x_0, y_0 \leq 10^{16}, 2 \leq a_x, a_y \leq 100, 0 \leq b_x, b_y \leq 10^{16}, 1 \leq x_s, y_s, t \leq 10^{16})$
- 题解:可以看出点的坐标是按照指数增长，因此需要考虑的点数量很少，不到100个。我们可以先枚举先从起点到哪个点，然后先往左走，如果到头则去走右边的点，这样一定能包含最优解。考虑先往左走走到头再往回走过程中白走的路，和先往右走到某个点再往回走中白走的路，因为$a_x, a_y \ge 2$，因此前者更小，再加上左边点的密度大，所以这样一定是最优的。

E

- 题意:n个节点的树，将0到$n-2$填到每条边上，使得$\sum_{1 \leq u < v \leq n} mex(u, v)$最小，其中$mex(u, v)$表示$u$到$v$路径上所有边权值的$mex$值。$(2 \leq n \leq 3000)$
- 题解:可以发现按照从小到大去给边赋值，那么对答案有贡献的边是连续的，且构成一个单谷序列。设$siz_{i, j}$表示以i为根时，j为根的子树大小，$fa_{i, j}$表示以i为根时，j的父亲节点，$f_{i,j}$表示以i,j为有贡献序列的两端的最大答案。那么有$f_{i, j} = \max(f_{i, fa_{i, j}}, f_{fa_{j, i}, v}) + siz_{i, j} \times siz_{j,i}$，预处理和记忆化搜索都是$O(n^2)$。

F

- 题意:按如下方法构造一棵树，每个点有一个正整数权值，如果$a|b$且$\frac{b}{a}$是b的最小

质因子，则a与b之间有一条边。现在给定n个点，可能重复，每个点的权值都是阶乘，形如$k_i!$现在要求找一个点使得这个n个点到这个点的距离之和最小，重复选的点要算多次$(1 \le n \le 10^6, 0 \le k_i \le 5000)$

- 题解:首先以1为根，然后一步一步往下走寻找重心。如果某个子节点对应的子树内的特殊点的数量x满足$x \ge n-x$那么往下走可以保证答案不会变差。而且注意到对于某一个节点，如果存在这样的子节点，那么是它唯一的，如果不存在，选择该点所得到的答案就是最优的，因为我们每一步都是选择的最优方案。因此我们现在只需要知道如何获取某个子树中有多少个特殊点，可以看出如果a是b的后代，那么组成b的所有质因子也是a的质因子，而且它们是a最大的质因子，也就是说当前点所代表的质因子序列，所有符合条件的点的质因子序列的一个后缀，按照这个性质进行维护即可。

```
#include <bits/stdc++.h>
#define maxn 5086

using namespace std;

typedef long long ll;

int n;
int a[maxn], b[maxn];
int p[maxn], cnt;
bool tag[maxn];
int x;
int f[maxn][maxn], num[maxn], g[maxn];
ll ans;//debug:n有1e6 不开ll见祖宗

int main(){
    for(int i = 2;i <= 5000;i++){
        if(!tag[i]) p[++cnt] = i;
        for(int j = 1;j <= cnt && p[j] * i <= 5000;j++){
            tag[p[j] * i] = true;
            if(i % p[j] == 0) continue;
        }
    }
    for(int i = 2;i <= 5000;i++){
        x = i, memcpy(f[i], f[i - 1], sizeof(f[i - 1])), num[i] = num[i
- 1], g[i] = g[i - 1];
        for(int j = 1;x != 1 && j <= cnt;j++){
            while(x % p[j] == 0) f[i][j]++, num[i]++, g[i] = max(g[i],
j), x /= p[j];
        }
    }
    scanf("%d", &n);
    for(int i = 1;i <= n;i++) scanf("%d", &x), ++a[x];
    for(int i = 1;i <= 5000;i++){
        ans += 1ll * a[i] * num[i];
        b[g[i]] += a[i];
        //printf("%d %d--\n", a[i], g[i]);
    }
    //printf("%d--\n", ans);
```

```
    while(1){
        bool suc = false;
        for(int i = 1;i <= cnt;i++){
            if(b[i] >= n - b[i]){
                //printf("%d %d--\n", p[i], b[i]);
                suc = true;
                ans -= b[i] - (n - b[i]);
                for(int j = 1;j <= 5000;j++){
                    if(g[j]){
                        if(g[j] == i){
                            f[j][i]--;
                            if(!f[j][i]){
                                b[i] -= a[j];
                                while(!f[j][g[j]] && g[j]) --g[j];
                                if(g[j]) b[g[j]] += a[j];
                            }
                        }else{
                            b[g[j]] -= a[j], g[j] = 0;//debug:b[g[j]]打成b[i]
                        }
                    }
                }
                break;
            }
        }
        if(!suc) break;
    }
    printf("%lld", ans);
}
```