

# 2020.06.22-2020.06.28 周报

## 团队训练

2020.06.24 [2017-2018 ACM-ICPC, NEERC, Northern Subregional Contest](#) prob:8:8:14 rnk:11/?

[20200624比赛记录](#)

## \_wzx27

### 容斥原理

[weekly8](#)

给一个集合  $A$  问  $[1,N]$  有多少个数满足  $A$  中的元素都不是它的因数。

比较模板的一个题目，二进制枚举  $A$  的子集，根据集合大小的奇偶性容斥。

```
#include<bits/stdc++.h>
#define ll long long
#define pii_ pair<int,int>
#define mp_ make_pair
#define pb push_back
#define fi first
#define se second
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define per(i,a,b) for(int i=(a);i>=(b);i--)
#define show1(a) cout<<#a<<" = "<<a<<endl
#define show2(a,b) cout<<#a<<" = "<<a<<" "; cout<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
const int maxn = 2e5+5;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}

ll L,R; int n,a[25];

int main()
{
    fastio();
    cin>>L>>R>>n;
    rep(i,0,n-1){
        cin>>a[i];
    }
    ll ans = 0;
    rep(i,0,((1<<n)-1)){
```

```
ll now = 1; int cnt = 0, flag = 0;
rep(j, 0, n-1){
    if((i>>j)&1){
        cnt++;
        if(now<=R/a[j]) now = now*a[j];
        else flag=1;
    }
}
ll tmp;
if(flag) tmp = R-L+1;
else tmp = R-L+1 - (R/now - (L-1)/now);
if(cnt&1) ans += tmp;
else ans -= tmp;
}
cout<<ans<<endl;
return 0;
}
```

## P5628

给一颗树，每条边的边权为它分割出的两个子树大小的乘积。对去掉某个点以及和这个点距离不超过k的其他点之后，损失的最大边权。

任取根，边权可以一次  $\text{dfs}$  求出每个点的子树大小，也就可以求出他的父亲边的边权。

然后考虑树形  $\text{dp}$   $f[u][j]$  表示去掉点  $u$  以及和它距离不超过  $j$  的其他点之后损失的边权。那么有  $f[u][j] = \sum_{v \in \text{son}(u)} f[v][j-1] - (\text{deg}[u]-1) \times f[u][j-2]$

```
#include<bits/stdc++.h>
#define ll long long
#define pii_ pair<int,int>
#define mp_ make_pair
#define pb push_back
#define fi first
#define se second
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define per(i,a,b) for(int i=(a);i>=(b);i--)
#define show1(a) cout<<#a<<" = "<<a<<endl
#define show2(a,b) cout<<#a<<" = "<<a<<" "; cout<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
const int maxn = 3e4+5;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}

int n,k,sz[maxn],deg[maxn],father[maxn];
vector<int> g[maxn];
ll f[maxn][205];
```

```

void dfs(int u,int fa)
{
    sz[u] = 1; father[u] = fa;
    for(int v:g[u])if(v!=fa){
        dfs(v,u);
        sz[u] += sz[v];
        f[u][0] += (ll)sz[v] * (n-sz[v]);
        f[v][0] += (ll)sz[v] * (n-sz[v]);
    }
}
int main()
{
    fastio();
    cin>>n>>k;
    rep(i,1,n-1){
        int u,v; cin>>u>>v;
        g[u].pb(v),g[v].pb(u);
        deg[u]++,deg[v]++;
    }
    dfs(1,0);
    ll ans = 0;
    rep(j,1,k){
        rep(i,1,n){
            if(j>1) f[i][j] = f[i][j-2];
            for(int v:g[i]){
                f[i][j] += f[v][j-1];
                if(j>1) f[i][j] -= f[i][j-2];
            }
        }
    }
    rep(i,1,n){
        ans = max(ans,f[i][k]);
    }
    cout<<ans<<endl;
    return 0;
}

```

## P1450

有四种币值的硬币  $c_i$  有  $t$  组询问，每组有  $s$  个整数  $d_i$  和  $s$  要求有  $d_i$  个  $c_i$  的硬币的情况下，恰好组合出  $s$  的方案数。

如果没有个数的限制就是完全背包，有个数的限制则考虑容斥，减掉一种硬币不满足的个数，加上两种硬币不满足的个数....

求解的时候先  $\text{dp}$  求出在完全背包的情况下组合出  $s$  的方案数  $f[i]$  然后用  $\text{dfs}$  模拟容斥的过程。

```

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

```

```
#define pii_ pair<int,int>
#define mp_ make_pair
#define pb push_back
#define fi first
#define se second
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define per(i,a,b) for(int i=(a);i>=(b);i--)
#define show1(a) cout<<#a<<" = "<<a<<endl
#define show2(a,b) cout<<#a<<" = "<<a<<" "<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
const int maxn = 2e5+5;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}

ll f[maxn],ans;
int a[5],b[5];

void dfs(int id,int num,ll s)
{
    if(s<0) return ;
    if(id==5){
        if(num&1) ans -= f[s];
        else ans += f[s];
        return ;
    }
    dfs(id+1,num,s);
    dfs(id+1,num+1,s-((ll)b[id]+1)*a[id]);
}

int main()
{
    fastio();
    rep(i,1,4) cin>>a[i];
    int t ;cin>>t;
    f[0] = 1;
    rep(i,1,4){
        rep(j,1,100000) if(j>=a[i]) f[j]+=f[j-a[i]];
    }
    while(t--){
        rep(i,1,4) cin>>b[i]; int s;cin>>s;
        ans = 0;
        dfs(1,0,s);
        cout<<ans<<endl;
    }
    return 0;
}
```

## Infinity37

### 专题

### 题目

[B.Boolean Satisfiability](#)

[C.Consonant Fency](#)

[I.Intelligence in Perpendicularia](#)

### 比赛

[Codeforces Round 652 \(Div. 2\) Infinity37比赛记录](#)

[Educational Codeforces Round 90 Infinity37比赛记录](#)

## Zars19

\摸= =摸/

### 比赛

[Codeforces Round 651 Div. 2 Zars19 \*\*DONE\*\*](#)

[Codeforces Round 652 Div. 2 Zars19](#)

## 本周推荐

From:

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

Permanent link:

[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai\\_milk:weekly8&rev=1593843020](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai_milk:weekly8&rev=1593843020)



Last update: **2020/07/04 14:10**