CF660 div2

A

题意：给定一个正整数，问是否能将其表示成四个不同正整数的和，使得其中至少有三个正整数都可以写成两个不同素数乘积的形式。

题解：水题，直接取最小的几个可以写成不同素数乘积的数字：6,10,14，即可。要特判最后一个数是否与前面不同，如果相同，把14换成15即可。

```
#include <algorithm>
#include <cmath>
#include <cstdio>
#include <cstring>
#include <iostream>
using namespace std;
int main()
{
    int t;
    cin >> t;
    while(t--)
    {
        int n;
        cin >> n;
        if (n > 30)
        {
            int tem = n - 30;
            if (tem == 10)
                cout << "Yes"<<endl<<"1 10 14 15" << endl;
            else if (tem == 6)
                cout << "Yes"<<endl<<"5 6 10 15" << endl;
            else if (tem == 14)
                cout << "Yes"<<endl<<"6 7 10 21" << endl;
            else
            {
                cout << "Yes" << endl;
                cout << 6 << " " << 10 << " " << 14 << ' ' << n - 30 <<
endl;
            }
        }
        else
            cout << "No" << endl;
    }
    return 0;
}
```

B

题意：给定n□要求写一个n位的十进制数字，将其转换成二进制后，擦去这个二进制数字的最后n位，要求在保证擦除后二进制数字最大的情况下，这个十进制数字尽量小。

题解：发现擦除后，最后的数字是9还是8都不影响二进制数字的大小，但要求十进制数字尽量小，所以选择8，所以最后构造的数字就是99999…..88888.

```
#include <algorithm>
#include <cmath>
#include <cstdio>
#include <cstring>
#include <iostream>
using namespace std;
int main()
{
    int t;
    cin >> t;
    while(t--)
    {
        int n;
        cin >> n;
        int tem = n / 4;
        if (n % 4 != 0)
            tem++;
        int i;
        for (i = 1; i <= n - tem;i++)
            putchar('9');
        for (; i <= n;i++)
            putchar('8');
        putchar('\n');
    }
    return 0;
}
```

C

题意：给定一棵树，刚开始有m个人都在根节点，要往他们的各自的目标节点走，给出以各个节点为目标的人数P_i,以及，每个人都有好心情或者坏心情，在中途一个人的心情可以由好变坏，但不可以由坏变好，给定每个节点经过的所有人的经过节点时的好心情人数与坏心情人数之差H_i□问这个H_i是否合法，即能否存在一种情况，使得每个节点的H_i为给定数字。

题解：DFS□可以列方程根据经过节点人数以及好坏心情人数之差算出经过该节点的好心情人数与坏心情人数，由于心情只能变坏，所以比较该节点的好心情人数与其子节点的好心情人数，不能变多，变多即不存在，或者方程的解并非整数，也即不存在。

```
#include <algorithm>
```

```cpp
#include <cmath>
#include <cstdio>
#include <cstring>
#include <iostream>
#include <stdio.h>
using namespace std;
const int MAXN = 1e5 + 20;
const int MAXM = 2e5 + 40;
int head[MAXN], to[MAXM], nex[MAXM], tot;
int sumpeo[MAXN],selflaf[MAXN],selfcry[MAXN],totlaf[MAXN],totcry[MAXN];
int P[MAXN], H[MAXN];
int n, m,flag = 0;
void addedge(int x,int y)
{
    tot++;
    to[tot] = y;
    nex[tot] = head[x];
    head[x] = tot;
}
void init()
{
    memset(head, 0, sizeof(head));
    tot = 0;
    flag = 0;
}
void dfs(int u,int fa)
{
    int temlaf = 0, temcry = 0;
    sumpeo[u] = P[u];
    for (int i = head[u]; i;i = nex[i])
    {
        if (to[i] == fa)
            continue;
        else
            dfs(to[i], u);
        temlaf += totlaf[to[i]];
        temcry += totcry[to[i]];
        sumpeo[u] += sumpeo[to[i]];
    }
    int doulaf = H[u] + sumpeo[u];
    if (doulaf < 0 || doulaf %2 != 0)
        flag = 1;
    totlaf[u] = doulaf / 2;
    int doucry = sumpeo[u] - H[u];
    if (doucry < 0 || doucry %2 != 0)
        flag = 1;
    totcry[u] = doucry / 2;
    if (temlaf > totlaf[u])
        flag = 1;
}
int main()
```

```
{
    int t;
    cin >> t;
    while(t--)
    {
        cin >> n >> m;
        init();
        for (int i = 1; i <= n;i++)
            scanf("%d", &P[i]);
        for (int i = 1; i <= n;i++)
            scanf("%d", &H[i]);
        for (int i = 1; i <= n - 1;i++)
        {
            int x, y;
            scanf("%d%d", &x, &y);
            addedge(x, y);
            addedge(y, x);
        }
        dfs(1, -1);
        if (!flag)
            puts("Yes");
        else
            puts("No");

    }
    return 0;
}
```

D

题意：略

题解：将b_i作为边建图，题目保证是DAG□直接从$b_i = -1$的点开始dfs□对于$a_i< 0$的点，要存进一个栈□$a_i>0$的点存进另一个栈，对于一个节点，可以选完$val > 0$的子节点再选它。最终输出答案的时候注意对小于0的栈倒序输出。

```
#include <algorithm>
#include <cmath>
#include <cstdio>
#include <cstring>
#include <iostream>
using namespace std;
typedef long long ll;
const int N=2e5+5,M=2e4+5;
#define mst(a) memset(a,0,sizeof a)
ll a[N],ans;
int n,m,b[N];
vector<int>c,d,e[N];
```

```
void dfs(int u,int fa){
    for(auto v:e[u]){
        if(v==fa) continue;
        dfs(v,u);
        if(a[v]>=0) a[u]+=a[v];
    }
    a[u]>=0?c.push_back(u):d.push_back(u);
    ans+=a[u];
}
int main(){
    scanf("%d",&n);
    for(int i=1;i<=n;i++)
        scanf("%lld",&a[i]);
    for(int i=1;i<=n;i++)
    {
        scanf("%d",&b[i]);
        if(~b[i])
            e[b[i]].push_back(i);
    }
    for(int i=1;i<=n;i++)
        if(b[i]==-1)
            dfs(i,0);
    printf("%lld\n",ans);
    for(int i:c)
        printf("%d ",i);
    for(int i=d.size()-1;~i;i--)
        printf("%d ",d[i]);
    return 0;
}
```