ABC_239

d.Prime Sum Game

Takahashi要赢,就要在a~b范围内找到数**x**
对于所有c~d内的任一数**y**都有:(x+y)为合数
否则Aoki赢

```
const int max_n = 2e2+10;
int a,b,c,d;
bool flag = 0;

namespace my_tem{
    bool flag_prime[max_n];
    int prime[max_n];

    int prime_list(int range)
    {
        int cnt=0;
        std::memset(flag_prime,true,sizeof(flag_prime));
        for(int i=2;i<=range;i++)
        {
            if(flag_prime[i])
            prime[++cnt]=i;   // begin:prime[1]
            for(int j=1;j<=cnt;j++){
                if(i*prime[j]>range)
                    break;
                flag_prime[i*prime[j]]=0;
                if(i%prime[j]==0)
                    break;
            }
        }
        return cnt;
    }
}

int main(){
    scanf(" %d %d %d %d",&a,&b,&c,&d);
    my_tem::prime_list(max_n);
    for(int i = a;i <= b;i++){
        flag = 1;
        for(int j = c;j <= d;j++){
            if(my_tem::flag_prime[i+j])
                flag = 0;
        }
        if(flag)
            break;
        else
            flag = 0;
    }
    printf("%s\n",flag ? "Takahashi" : "Aoki");
    return 0;
}
```

E.Subtree K-th Max

对于所有顶点,可以将以该点为树根的所有顶点排序记录下来
使用递归方式得到结果
直接查询即可

```
#include<iostream>
#include<algorithm>
#include<cstdio>
#include<cmath>
#include<deque>
#include<queue>

const int max_n = 1e5+10;
int ver[max_n];
int N,Q;
int t1,t2;
std::vector<int> edge[max_n],nums[max_n];

struct edge{
    int from,to;
};

void dfs(int v,int father){
    nums[v].push_back(ver[v]);
    for(auto it : edge[v]){
        if(it == father)
            continue;
        dfs(it,v);
        for(auto i : nums[it])
            nums[v].push_back(i);
    }
    std::sort(nums[v].begin(),nums[v].end(),std::greater<int>());
    while(nums[v].size() > 20)
        nums[v].pop_back();
}

int main(){
    scanf(" %d %d",&N,&Q);
    for(int i = 1;i <= N;i++)
        scanf(" %d",&ver[i]);
    for(int i = 1;i <= N-1;i++){
        scanf(" %d %d",&t1,&t2);
        edge[t1].push_back(t2);
        edge[t2].push_back(t1);
    }
    dfs(1,0);
    while(Q--){
        scanf(" %d %d",&t1,&t2);
        printf("%d\n",nums[t1][t2-1]);
    }
    return 0;
}
```