

综合练习

人员

谢亚锴、隋天乙 到课

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/1030>

🏠 比赛概况

📋 题目列表

📋 选择题列表

📄 提交记录

★ 实时榜单

★ 选择题排行榜

王向东老师周日十点半C++三分

🔄 刷新

#	用户名	姓名	编程分	时间	A	B	C	D
1	sunjingke	孙靖轲	400	9292	100	100	100	100
2	yuxiaolong	于霄龙	300	1577	100	100	100	
3	liuchuangsuo	刘闯速	300	1639	100	100	100	
4	yangjinshuo	杨谨硕	200	722	100	100		
5	niutongze	牛同泽	200	1102	100	100		
6	yuzijia1	于子珈	150	452	100	50		
7	xieyakai	谢亚锴	100	316	100			
8	chenluoran	陈洛冉	100	323	100			
9	siyunxin	司云心	100	458	100	0		

本周作业

<https://cppoj.kids123code.com/contest/1116> (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

同学们课上听讲做题都很认真。

课堂内容

Tree and Hamilton Path 2

用 总边权*2 - 树的直径 即可

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
struct node {
    int to, val;
};
vector<node> vec[maxn];
```

```

LL d[maxn];

void dfs(int u, LL v) {
    if (d[u] != -1) return;

    d[u] = v;
    for (node it : vec[u]) dfs(it.to, v+it.val);
}

int main()
{
    int n; cin >> n;
    LL res = 0;
    for (int i = 1; i <= n-1; ++i) {
        int a, b, c; cin >> a >> b >> c; res += c*2;
        vec[a].push_back({b,c}), vec[b].push_back({a,c});
    }

    memset(d, -1, sizeof(d)); dfs(1, 0);
    int id = 1;
    for (int i = 2; i <= n; ++i) {
        if (d[i] > d[id]) id = i;
    }

    memset(d, -1, sizeof(d)); dfs(id, 0);
    LL maxx = 0;
    for (int i = 1; i <= n; ++i) maxx = max(maxx, d[i]);

    cout << res - maxx << endl;
    return 0;
}

```

Take ABC

类似于括号匹配, 用栈存未删除的字符, 当这个字符是C, 且前两个是B和A的时候, 删除; 否则, 把字符往栈里放

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;
char s[maxn];

int main()
{
    cin >> (s+1);
    int n = strlen(s+1);

    vector<char> vec; int len = 0;
    for (int i = 1; i <= n; ++i) {
        if (s[i]=='C' && len>=2 && vec[len-1]=='B' && vec[len-2]=='A') {

```

```

        vec.pop_back(), vec.pop_back(), len -= 2;
    } else vec.push_back(s[i]), ++len;
}

for (char i : vec) cout << i;
cout << endl;
return 0;
}

```

Strange Lunchbox

dp

$f[i][j]$ 代表, 当拥有 $\geq i$ 个章鱼烧, 拥有 $\geq j$ 个大饼时, 最少用多少饭盒

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 600 + 5;
const int inf = 0x3f3f3f3f;
int a[maxn], b[maxn];
int f[maxn][maxn]; // f[i][j]:  $\geq i$  章/ $\geq j$  太 时 -> 最少多少个饭盒

int main()
{
    int n, x, y; cin >> n >> x >> y;
    for (int i = 1; i <= n; ++i) cin >> a[i] >> b[i];

    memset(f, 0x3f, sizeof(f)); f[0][0] = 0;
    for (int i = 1; i <= n; ++i) {
        for (int j = x; j >= 0; --j) {
            for (int k = y; k >= 0; --k) {
                f[j][k] = min(f[j][k], f[max(j-a[i], 0)][max(k-b[i], 0)]+1);
            }
        }
    }

    cout << (f[x][y]==inf ? -1 : f[x][y]) << endl;
    return 0;
}

```

Family and Insurance

输入时, 先把结果存到 f 数组中

$f[i]$: 在第 i 个人上最多往后传多少代

最后做一遍 dfs 搜索, 记录哪些点会被保险即可

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 3e5 + 5;
vector<int> vec[maxn];
int f[maxn], res = 0;

void dfs(int u, int cnt) {
    cnt = max(cnt, f[u]);
    if (cnt) ++res;
    for (int i : vec[u]) dfs(i, max(0, cnt-1));
}

int main()
{
    int n, m; cin >> n >> m;
    for (int i = 2; i <= n; ++i) {
        int x; cin >> x; vec[x].push_back(i);
    }
    while (m -- ) { int a, b; cin >> a >> b; f[a] = max(f[a], b+1); }

    dfs(1, 0);

    cout << res << endl;
    return 0;
}
```