

# 分组背包

---

## 人员

---

左子毅、郑伟、朱奕鸣、杨洋、刘子淇、赵清航、周子航 到课

## 作业检查

---

左子毅 完成

郑伟 未完成

朱奕鸣 完成

杨洋 完成

刘子淇 完成

赵清航 完成

周子航 完成

## 作业

---

<https://www.luogu.com.cn/contest/161669>

C、D、E 三个题必做

## 课堂表现

---

周子航、刘子淇 2位同学课上可以很快的完成上节课留的必做题，提出表扬！！

除了周子航、刘子淇同学之外的其他5位同学，都不能较快的完成上节课布置的必做题，以后要在课后多加练习。

## 课堂内容

---

P1757 通天之分组背包

### 分组背包模板题

设一组中有  $n$  个物品，而且这  $n$  个物品相互冲突

那么这  $n$  个物品的转移过程都应该从上一组最后的状态转移而来

这样，这一组的物品才不会同时选到，才可以达到在这一组中只取一个物品的要求

核心代码：

```
memcpy(p, f, sizeof(p));
for (int i = 1; i <= n; ++i) {
    for (int j = m; j >= v[i]; --j) f[j] = max(f[j], p[j-v[i]] + w[i]);
}
```

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

using namespace std;

const int N = 1000 + 5, M = 100 + 5;
struct node {
    int v, c;
};
vector<node> vec[M];
int f[N], p[N];

int main()
{
    int m, n; cin >> m >> n;
    for (int i = 1; i <= n; ++i) {
        int a, b, c; cin >> a >> b >> c;
        vec[c].push_back({a, b});
    }

    for (int i = 1; i <= 100; ++i) {
        for (node it : vec[i]) {
            for (int j = m; j >= it.v; j--) {
                f[j] = max(f[j], p[j-it.v]+it.c);
            }
        }
        for (int j = 0; j < N; ++j) p[j] = f[j];
    }
    cout << f[m] << endl;
    return 0;
}
```

## CF1741E Sending a Sequence Over the Network

一个数有三种情况：

1. 作为左边序列的长度
2. 作为右边序列的长度
3. 作为序列中的数

通过分析，易知 一个数作为序列中的数 对最后的结果是没有贡献的，不做分析

分析情况1，作为左边序列的长度，应该是 `if (i-a[i]-1>=0) f[i] |= f[i-a[i]-1];`

分析情况2，作为右边序列的长度，应该是 `if (i+a[i]<=n) f[i+a[i]] |= f[i-1];`

T组测试数据，每次处理时记得清空数组

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

using namespace std;

const int maxn = 2e5 + 5;
int w[maxn];
bool f[maxn];

void solve() {
    int n; cin >> n;
    for (int i = 0; i <= n+2; ++i) w[i] = 0, f[i] = false;

    f[0] = true;
    for (int i = 1; i <= n; ++i) {
        cin >> w[i];
        if (i-w[i]-1 >= 0) f[i] |= f[i-w[i]-1];
        if (i+w[i] <= n) f[i+w[i]] |= f[i-1];
    }
    if (f[n]) cout << "YES" << endl;
    else cout << "NO" << endl;
}

int main()
{
    int T; cin >> T;
    while (T -- ) { solve(); }
    return 0;
}
```

## CF1286B Numbers on Tree

1. 判断是否存在一个 $u$ 节点，它的孩子节点个数超过 $c[u]$ 。如果存在，输出NO；否则，输出YES
2. 从根节点往下遍历，若遍历到一个点 $u$ ，则找到第 $c[u]$ 个未标记的数即可

核心代码：

```
int cnt = 0;
for (int i = 1; i <= n; ++i) {
    if (!vis[i]) {
        if (cnt == c[u]) {
            vis[i] = true;
            ans[u] = i;
            break;
        }
        ++cnt;
    }
}
```

此题为课下思考作业题，不提供具体代码