

# 树上背包

---

## 人员

---

于潇涵、周子航、高可维、杨青悠、王梓同、蔡云翔、石宇焮、李佳声、王崇宇、窦浩轩 到课

## 作业检查

---

于潇涵 完成

周子航 完成

高可维 未完成

杨青悠 未完成

王梓同 未完成

蔡云翔 未完成

石宇焮 完成

李佳声 未完成

王崇宇 未完成

窦浩轩 未完成

## 作业

---

<https://www.luogu.com.cn/contest/162596#problems>

A、B 两个题必做，B题要求用分组背包做法做一遍

<https://www.luogu.com.cn/contest/162467#problems>

这个刷题题单要求同学们每一道题都可以在限制时间内完成

## 课堂表现

---

课堂上同学们都不爱回答问题，大家以后上课要更加主动一些

# 课堂内容

## P5322 [BJOI2019] 排兵布阵

$h[i][j]$ : 第 $i$ 个城堡, 如果有 $j$ 个人, 能得到多少分 ( $j$ 的取值:  $0 \sim 20000$ , 最多 $s$ 个有意义)

现在一共有 $m$ 个人, 把它们分到第一个城堡, 第二个城堡, ...

假如说有4个城堡, 可以把这4个城堡当成4个组:

第一个城堡只能去0/1/2/3/4个人, 这5种方案只能选一种

对于第二、三、四个城堡同理

-> 分组背包: 每个组里只有选一种物品

$f[i]$ : 如果有 $i$ 个士兵, 最多得多少分

-> 答案:  $f[m]$

```
for (int i = 1; i <= n; ++i) {
    memcpy(p, f, sizeof(p));
    for (int j = 0; j <= m; ++j) { // 这一维可以从O(m)优化到O(s)
        for (int k = m; k >= j; --k) {
            f[k] = max(f[k], p[k-j] + h[i][j]);
        }
    }
}
```

```

#include <bits/stdc++.h>

using namespace std;

const int N = 100 + 5, M = 20000 + 5;
vector<int> vec[N];
int f[M];

int main()
{
    int s, n, m; cin >> s >> n >> m;
    for (int i = 1; i <= s; ++i) {
        for (int j = 1; j <= n; ++j) {
            int x; cin >> x; vec[j].push_back(x);
        }
    }
    for (int i = 1; i <= n; ++i) {
        vec[i].push_back(0);
        sort(vec[i].begin(), vec[i].end());
        for (int k = m; k >= 0; --k) {
            for (int j = 1; j <= s; ++j) {
                int t = 2*vec[i][j] + 1;
                if (k < t) continue;
                f[k] = max(f[k], f[k-t] + j*i);
            }
        }
    }
    cout << f[m] << endl;
    return 0;
}

```

## P2015 二叉苹果树

// 方法一：

$f[u][k]$ ：以 $u$ 为根，保留 $k$ 根树枝时最多能有多少苹果

设 $u$ 有2个孩子 $v1$ 、 $v2$ ，边权分别为 $a1$ 、 $a2$

$$f[u][k] = \max($$

$$f[v1][k-1] + a1$$

$$f[v1][k-2] + a1 + a2 + f[v2][0]$$

$$f[v1][k-3] + a1 + a2 + f[v2][1]$$

$$\dots\dots$$

$$f[v1][0] + a1 + a2 + f[v2][k-2]$$

$$a2 + f[v2][k-1]$$

$$)$$

该方法实现比较简单，不提供具体代码

// 方法二:

$f[u][k]$ : 以 $u$ 为根, 保留 $k$ 根树枝时最多能有多少苹果

设 $u$ 有5个孩子 $v_1$ 、 $v_2$ 、 $v_3$ 、 $v_4$ 、 $v_5$ , 边权分别为 $a_1$ 、 $a_2$ 、 $a_3$ 、 $a_4$ 、 $a_5$

那么接下来对于第一个孩子 $v_1$ , 需要在 $f[v_1][0]$ ,  $f[v_1][1]$ ,  $f[v_1][2]$ ,  $f[v_1][3]$ , ... 里选择一个方案

对于第二个孩子 $v_2$ , 需要在 $f[v_2][0]$ ,  $f[v_2][1]$ ,  $f[v_2][2]$ ,  $f[v_2][3]$ , ... 里选择一个方案

对于 $v_3$ ,  $v_4$ ,  $v_5$ 来说同理

最终, 希望选 $k$ 根树枝时能保留的苹果最多, 则这个问题转化为一个 分组背包 问题

// 方法二代码:

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

```
using namespace std;
```

```
const int maxn = 100 + 5;
```

```
struct node {
```

```
    int to, value;
```

```
};
```

```
vector<node> vec[maxn];
```

```
int f[maxn][maxn]; // f[i][j]: 第i个点, 保留j根树枝
```

```
void dfs(int u, int fa, int m) {
```

```
    for (node it : vec[u]) {
```

```
        if (it.to == fa) continue;
```

```
        dfs(it.to, u, m);
```

```
        for (int i = m; i >= 0; --i) {
```

```
            for (int j = 0; j <= i-1; ++j) {
```

```
                f[u][i] = max(f[u][i], f[u][i-j-1] + f[it.to][j] + it.value);
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    int n, m; cin >> n >> m;
```

```
    for (int i = 1; i <= n-1; ++i) {
```

```
        int u, v, w; cin >> u >> v >> w;
```

```
        vec[u].push_back({v, w}), vec[v].push_back({u, w});
```

```
    }
```

```
    dfs(1, -1, m);
```

```
    cout << f[1][m] << endl;
```

```
    return 0;
```

```
}
```

## P1757 通天之分组背包

```

// 不需要额外数组的代码写法
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 1000 + 5;
LL f[maxn];

struct node {
    int v, c;
};
vector<node> vec[105];

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

```