

dp练习2

人员

于家瑞、洪晨棋、洪晨栋、于霄龙、郭栩睿、崔宸赫 到课, 陶汇笙、王恩泽 线上

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/1684>

The screenshot shows a competition results page with a navigation bar at the top. The main content is a table titled "王向东老师周六十点半C++dp练习2". The table has columns for rank (#), username, name, programming score (编程分), time (时间), and four specific problem scores (A, B, C, D). The data is as follows:

#	用户名	姓名	编程分	时间	A	B	C	D
1	yujiarui	于家瑞	400	1215	100	100	100	100
2	taohuishing	陶汇笙	300	958	100	100	100	
3	guoxurui	郭栩睿	205	339	100	100	5	
4	yuxiaolong	于霄龙	200	338	100	100		
5	hongchendong	洪晨栋	190	223	100	90		
6	wangenze	王恩泽	100	115	100			

本周作业

<http://cppoj.kids123code.com/contest/1784> (课上讲了 A ~ E 这些题, 课后作业是 F 题)

课堂表现

今天讲的 01 背包的内容, 01 背包可以说是 dp 中最重要、最基本的内容, 同学们课下一定要好好复习一下课上的几道题, 把这几道题掌握熟练

课堂内容

[HNOI2004] 打鼹鼠 (上周作业)

类似于最长上升子序列的做法, $f[i]$ 代表以 i 结尾的时候, 最多能打多少个鼹鼠, 从前面找一个最好的转过来即可

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

using namespace std;

const int maxn = 1e4 + 5;
vector<int> vec[maxn];
struct node {
    int t, x, y;
} w[maxn];
bool cmp(node p, node q) { return p.t < q.t; }
int f[maxn];
```

```

bool check(node p, node q) {
    int dis = abs(p.x-q.x) + abs(p.y-q.y);
    return q.t-p.t >= dis;
}

int main()
{
    int n, m; cin >> n >> m;
    for (int i = 1; i <= m; ++i) cin >> w[i].t >> w[i].x >> w[i].y;
    sort(w+1, w+m+1, cmp);

    int res = 0;
    for (int i = 1; i <= m; ++i) {
        f[i] = 1;
        for (int j = 1; j < i; ++j) {
            if (check(w[j], w[i])) f[i] = max(f[i], f[j]+1);
        }
        res = max(res, f[i]);
    }
    cout << res << endl;
    return 0;
}

```

【深基12.例1】部分背包问题

直接贪心，按照单位价值进行排序，优先选单位价值大的金币

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 100 + 5;
struct node {
    int m, v;
    double d_v;
} a[maxn];
bool cmp(node p, node q) {
    return p.d_v > q.d_v;
}

int main()
{
    int n, T; cin >> n >> T;
    for (int i = 1; i <= n; i++) {
        cin >> a[i].m >> a[i].v;
        a[i].d_v = 1.0 * a[i].v / a[i].m;
    }

    sort(a+1, a+n+1, cmp);
    double res = 0;
    for (int i = 1; i <= n; i++) {

```

```

        if (a[i].m <= T) {
            res += a[i].v;
            T -= a[i].m;
        } else {
            res += T * a[i].d_v;
            break;
        }
    }

    printf("%.2lf\n", res);
    return 0;
}

```

[NOIP 2005 普及组] 采药

$f[i][j]$: 只考虑前 i 项, 当背包体积为 j 时, 最多能获得的价值是多少

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1000 + 5;
int f[105][maxn];

int main()
{
    int T, n; cin >> T >> n;
    for (int i = 1; i <= n; ++i) {
        int t, v; cin >> t >> v;
        for (int j = 0; j <= T; ++j) {
            f[i][j] = f[i-1][j];
            if (j >= t) f[i][j] = max(f[i][j], f[i-1][j-t]+v);
        }
    }
    cout << f[n][T] << endl;
    return 0;
}

```

滚动数组优化, 压一维, $f[i]$: 当背包体积为 i 时, 最多能获得多少价值

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1000 + 5;
int f[maxn];

int main()
{

```

```

int T, n; cin >> T >> n;
for (int i = 1; i <= n; ++i) {
    int t, v; cin >> t >> v;
    for (int j = T; j >= t; --j) f[j] = max(f[j], f[j-t]+v);
}
cout << f[T] << endl;
return 0;
}

```

[USACO07DEC] Charm Bracelet S

$f[i]$: 当背包体积为 i 时, 能获得的最大价值是多少

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 12880 + 5;
int f[maxn];

int main()
{
    int n, m; cin >> n >> m;
    while (n -- ) {
        int c, v; cin >> c >> v;
        for (int j = m; j >= c; --j) f[j] = max(f[j], f[j-c]+v);
    }
    cout << f[m] << endl;
    return 0;
}

```

[NOIP 2001 普及组] 装箱问题

当做背包题来做, 看做每个物品的 体积 和 价值 都是 c , 可以认为在背包体积为 V 时, 最多能获得的最大价值是多少

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 20000 + 5;
int f[maxn];

int main()
{
    int V, n; cin >> V >> n;
    while (n -- ) {
        int c; cin >> c;
        for (int i = V; i >= c; --i) f[i] = max(f[i], f[i-c]+c);
    }
}

```

```
    }
    cout << V-f[V] << endl;
    return 0;
}
```

[NOIP 2006 普及组] 开心的金明

f[i]: 花 i 元钱时, 最多获得多少价值

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

using namespace std;

const int maxn = 30000 + 5;
int f[maxn];

int main()
{
    int n, m; cin >> n >> m;
    while (m -- ) {
        int v, p; cin >> v >> p;
        for (int i = n; i >= v; --i) f[i] = max(f[i], f[i-v]+p*v);
    }
    cout << f[n] << endl;
    return 0;
}
```