

# 综合练习

## 人员

陶汇笙、郭栩睿、崔宸赫、王恩泽、于霄龙、于家瑞 到课

## 上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/2450>

王向东老师周六十点半C++最长上升子序列										
#	用户名	姓名	编程分	时间	A	B	C	D	E	刷新
1	yujiarui	于家瑞	500	582	100	100	100	100	100	
2	yuxiaolong	于霄龙	500	1083	100	100	100	100	100	
3	cuchenhe	崔宸赫	500	1103	100	100	100	100	100	
4	taohuishing	陶汇笙	500	1128	100	100	100	100	100	
5	guoxurui	郭栩睿	430	1041	100	100	100	100	30	
6	hongchendong	洪晨栋	400	1043	100	100	100	100	100	
7	wangenze	王恩泽	400	1056	100	100	100	100	100	
8	hongchenqi	洪晨棋	400	1059	100	100	100	100	100	

## 本周作业

<https://cppoj.kids123code.com/contest/2580> (课上讲了 A ~ E 这些题, 课后作业是 E 题)

## 课堂表现

今天的 A、B 题非常基础, 有几位同学课上一开始没有做出来, 说明之前的完全背包基础不扎实, 需要课下多复习一下。

## 课堂内容

### [NOIP 2012 普及组] 摆花 (上周作业)

$f[i][j]$ : 只考虑前  $i$  种花, 一共买  $j$  盆时, 有多少方案

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

using namespace std;

const int maxn = 100 + 5;
const int mod = 1e6 + 7;
int f[maxn][maxn];

int main()
{
```

```

int n, m; cin >> n >> m;
f[0][0] = 1;
for (int i = 1; i <= n; ++i) {
    int x; cin >> x;
    for (int j = 0; j <= m; ++j) {
        for (int k = 0; k <= min(x,j); ++k) f[i][j] = (f[i][j]+f[i-1][j-k])%mod;
    }
}
cout << f[n][m] << endl;
return 0;
}

```

## 纸币问题 1

f[i]: 凑 i 元最少用几张纸币

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e4 + 5;
int f[maxn];

int main()
{
    int n, m; cin >> n >> m;
    memset(f, 0x3f, sizeof(f)); f[0] = 0;
    while (n-- ) {
        int x; cin >> x;
        for (int i = x; i <= m; ++i) f[i] = min(f[i], f[i-x]+1);
    }
    cout << f[m] << endl;
    return 0;
}

```

## 纸币问题 3

f[i]: 凑 i 元有多少方案, 用完全背包的方法做

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e4 + 5;
const int mod = 1e9 + 7;
int f[maxn];

int main()
{

```

```

int n, m; cin >> n >> m;
f[0] = 1;
for (int i = 1; i <= n; ++i) {
    int x; cin >> x;
    for (int j = x; j <= m; ++j) f[j] = (f[j] + f[j-x]) % mod;
}
cout << f[m] << endl;
return 0;
}

```

## 纸币问题 2

因为考虑了不同的顺序, 所以需要改变先后枚举的顺序

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e4 + 5;
const int mod = 1e9 + 7;
int w[maxn], f[maxn];

int main()
{
    int n, m; cin >> n >> m;
    for (int i = 1; i <= n; ++i) cin >> w[i];

    f[0] = 1;
    for (int i = 1; i <= m; ++i) {
        for (int j = 1; j <= n; ++j) {
            if (i >= w[j]) f[i] = (f[i] + f[i-w[j]]) % mod;
        }
    }
    cout << f[m] << endl;
    return 0;
}

```

## [CSP-X2025 山东] 勇者斗恶龙

每个数只会改变 0, 1, 2 次这 3 种情况, 因此可以用 dp 来做

```

#include <bits/stdc++.h>
#define int long long

using namespace std;

const int maxn = 2e5 + 5;
const int inf = 0x3f3f3f3f3f3f3f3f;
int a[maxn], b[maxn], f[maxn][3];

```

```

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

    memset(f, 0x3f, sizeof(f));
    f[1][0] = 0, f[1][1] = b[1], f[1][2] = b[1]*2;
    for (int i = 2; i <= n; ++i) {
        for (int j = 0; j <= 2; ++j) {
            for (int k = 0; k <= 2; ++k) {
                if (a[i-1]+k != a[i]+j) f[i][j] = min(f[i][j], f[i-1][k]+b[i]*j);
            }
        }
    }
    cout << min({f[n][0], f[n][1], f[n][2]}) << endl;
    return 0;
}

```

## [TJOI2007] 线段

这道题有些复杂

$f[i][0/1]$ : 处理到第  $i$  行, 以 左端点/右端点 结尾时, 最少走多少步

$f[i][0/1]$  可以从  $f[i-1][0/1]$  转移而来

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e4 + 5;
int f[maxn][2];
struct node {
    int l, r;
} w[maxn];

int calc(int st, int l, int r, bool is_right) {
    if (is_right) {
        if (st <= l) return r - st;
        return st - l + r - l;
    }
    if (st >= r) return st - l;
    return r - st + r - l;
}

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i].l >> w[i].r;
}

```

```
f[1][0] = 2*w[1].r - w[1].l - 1;
f[1][1] = w[1].r - 1;
for (int i = 2; i <= n; ++i) {
    int a = calc(w[i-1].l, w[i].l, w[i].r, true);
    int b = calc(w[i-1].r, w[i].l, w[i].r, true);
    f[i][1] = min(f[i-1][0]+a, f[i-1][1]+b);

    int c = calc(w[i-1].l, w[i].l, w[i].r, false);
    int d = calc(w[i-1].r, w[i].l, w[i].r, false);
    f[i][0] = min(f[i-1][0]+c, f[i-1][1]+d);
}

int res = min(f[n][0]+n-w[n].l, f[n][1]+n-w[n].r) + n-1;
cout << res << endl;
return 0;
}
```