

# 区间dp

## 人员

陶汇笙、郭栩睿、洪晨栋、洪晨棋、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/2235>

The screenshot shows a competition results page with a table of scores. The table has columns for # (rank), 用户名 (username), 姓名 (name), 编程分 (programming score), 时间 (time), A, B, C, and D. The data is as follows:

#	用户名	姓名	编程分	时间	A	B	C	D
1	wangenze	王恩泽	400	212	100	100	100	100
2	yuxiaolong	于霄龙	400	232	100	100	100	100
3	hongchenqi	洪晨棋	400	340	100	100	100	100
4	yujiarui	于家瑞	400	3645	100	100	100	100
5	hongchendong	洪晨栋	360	409	100	100	100	60
6	taohuisheng	陶汇笙	300	214	100	100		100
7	guoxurui	郭栩睿	300	249	100	100	100	0
8	cuchenhe	崔宸赫	100	39	100			

## 本周作业

<https://cppoj.kids123code.com/contest/2326> (课上讲了 A ~ D 这些题, 课后作业是 E 题)

## 课堂表现

今天学了区间 dp 这个内容, 课上 于霄龙、于家瑞、陶汇笙 几位同学思考很积极, 提出表扬!

## 课堂内容

### 榨取kkksc03 (上周作业)

#### 二维 01 背包

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

using namespace std;

const int maxn = 200 + 5;
int f[maxn][maxn];

int main()
{
    int n, M, T; cin >> n >> M >> T;
    while (n--) {
```

```

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

```

## [NOIP 2004 提高组] 合并果子

贪心, 每次取最小的两个合并

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    priority_queue<int, vector<int>, greater<int>>q;
    int n; cin >> n;
    while (n -- ) { int x; cin >> x; q.push(x); }

    int res = 0;
    while (q.size() > 1) {
        int t1 = q.top(); q.pop();
        int t2 = q.top(); q.pop();
        int t = t1 + t2;
        q.push(t); res += t;
    }
    cout << res << endl;
    return 0;
}

```

## 石子合并 (弱化版)

$f[i][j]$ : 把  $i \sim j$  合并时, 最小花费多少

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 300 + 5;
const int inf = 0x3f3f3f3f;
int w[maxn], f[maxn][maxn];
int p[maxn];

int sum(int l, int r) { return p[r] - p[l-1]; }

```

```

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

    for (int len = 2; len <= n; ++len) {
        for (int i = 1; i+len-1 <= n; ++i) {
            int j = i+len-1;
            f[i][j] = inf;
            for (int k = i; k < j; ++k)
                f[i][j] = min(f[i][j], f[i][k] + f[k+1][j] + sum(i,j));
        }
    }
    cout << f[1][n] << endl;
    return 0;
}

```

## [NOI1995] 石子合并

整体思路跟上个题一致，多一个破坏成链的思想，把数组扩两倍

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 200 + 5;
const int inf = 0x3f3f3f3f;
int w[maxn], p[maxn];
int f1[maxn][maxn], f2[maxn][maxn];

int sum(int l, int r) { return p[r] - p[l-1]; }

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i], w[n+i] = w[i];
    for (int i = 1; i <= 2*n; ++i) p[i] = p[i-1] + w[i];

    memset(f1, 0x3f, sizeof(f1));
    for (int i = 1; i <= 2*n; ++i) f1[i][i] = 0;

    for (int len = 2; len <= n; ++len) {
        for (int i = 1; i+len-1 <= 2*n; ++i) {
            int j = i+len-1;
            for (int k = i; k < j; ++k) {
                f1[i][j] = min(f1[i][j], f1[i][k] + f1[k+1][j] + sum(i,j));
                f2[i][j] = max(f2[i][j], f2[i][k] + f2[k+1][j] + sum(i,j));
            }
        }
    }
}

```

```

int minn = 1e9, maxx = 0;
for (int i = 1; i <= n; ++i) {
    minn = min(minn, f1[i][i+n-1]);
    maxx = max(maxx, f2[i][i+n-1]);
}
cout << minn << endl;
cout << maxx << endl;
return 0;
}

```

## [IOI 2000] 回文字符串

$f[i][j]$ : 把  $i \sim j$  变为回文时, 最少需要插入几个字符

当  $s[i]==s[j]$  时,  $f[i][j] = f[i+1][j-1]$

当  $s[i] \neq s[j]$  时,  $f[i][j] = \min(f[i+1][j], f[i][j-1]) + 1$

```

#include <bits/stdc++.h>

using namespace std;

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

int main()
{
    cin >> (s+1);
    int n = strlen(s+1);
    for (int len = 2; len <= n; ++len) {
        for (int i = 1; i+len-1 <= n; ++i) {
            int j = i+len-1;
            if (s[i] == s[j]) f[i][j] = f[i+1][j-1];
            else f[i][j] = min(f[i+1][j], f[i][j-1]) + 1;
        }
    }
    cout << f[1][n] << endl;
    return 0;
}

```