# 四边形不等式优化

主讲人：邓哲也

# 四边形不等式优化

四边形不等式优化主要针对区间DP模型。

转移方程形如：

$$f[i][j] = \min(f[i][k] + f[k+1][j] + w(i, j))$$

对于每个区间都要枚举 k

时间复杂度 $O(n^3)$

状态数已经定好了是 $O(n^2)$，我们能想办法优化的部分就是

把 $O(n)$ 的转移优化到 $O(1)$

这里就要用到四边形不等式优化。

# 四边形不等式

对于 $a < b < c < d$：

如果有 $f(a, c) + f(b, d) <= f(b, c) + f(a, d)$

我们就称 f 满足四边形不等式。

# 四边形不等式优化

如果代价函数 $w(i,j)$ 满足单调性和四边形不等式，那么 dp 函数 $f(i,j)$ 也满足四边形不等式。

定义 $s(i,j)$ 为 $f(i,j)$ 取得最优值对应的转移（即 k）

如果 $f(i,j)$ 满足四边形不等式，那么 $s(i,j)$ 单调，即

$s(i,j) \leqslant s(i,j+1) \leqslant s(i+1,j+1)$

# 合并果子

有 n 堆果子，第i堆有 a[i] 个。

合并的时候只能合并相邻两堆，产生的代价为两堆果子的总数。

经过 n-1 轮合并后变成了一堆，求总代价的最小值。

样例：（输出 9）

3

1 2 3

# 合并果子

状态：

用 f[i][j] 来表示合并 [i,j] 这个区间内的果子产生的最小代价。

转移：

f[i][j] = min(f[i][j], f[i][k] + f[k + 1][j] + sum[j] - sum[i - 1])

# 合并果子

这里w(i, j)=sum[j]-sum[i-1]，也就是区间 [i, j] 的果子总数。

显然满足四边形不等式:

$$w(a,c)+w(b,d) \leqslant w(b,c)+w(a,d)$$

# 合并果子

下面看 f 是否满足四边形不等式，也就是对于 $i<i+1\leqslant j<j+1$，要满足 $f[i][j]+f[i+1][j+1]\leqslant f[i+1][j]+f[i][j+1]$

令 $x=s[i][j+1]$，$y=s[i+1][j]$

把 x 和 y 带入，得到：

左式≤

$f[i][x]+f[x+1][j]+w(i,j)+f[i+1][y]+f[y+1][j+1]+w(i+1,j+1)$

因为 $w(i,j)+w(i+1,j+1)\leqslant w(i+1,j)+w(i,j+1)$

所以左式≤

$f[i][x]+f[x+1][j+1]+w(i+1,j)+f[i+1][y]+f[y+1][j]+w(i,j+1)$

$=f[i][j+1]+f[i+1][j]$

# 合并果子

最后我们证明 s[i][j-1]≤s[i][j]≤s[i+1][j]

对于 s[i][j-1]≤s[i][j]，设 y = s[i][j-1]，对于 x < y

x+1<y+1≤j-1<j

根据四边形不等式有：

f[x+1][j-1]+f[y+1][j] ≤ f[y+1][j-1]+f[x+1][j]

两边都加上 f[i][x]+w[i][j-1]+f[i][y]+w[i][j]

得到 (f[i][x]+f[x+1][j-1]+w[i][j-1]) +

(f[i][y]+f[y+1][j]+w[i][j]) ≤ (f[i][y]+f[y+1][j-1]+w[i][j-1])

+ (f[i][x]+f[x+1][j]+w[i][j])

即 f[i][j-1] + f[i][j] ≤ f[i][j-1] + f[i][j]

(k=x) (k=y) (k=y) (k=x)

# 合并果子

即 $f[i][j-1] + f[i][j] \leqslant f[i][j-1] + f[i][j]$

$\quad\quad(k=x)\quad\quad(k=y)\quad\quad\quad(k=y)\quad\quad(k=x)$

因为 $y = s[i][j-1]$，所以 $f[i][j-1](k=x)$ 一定大于 $f[i][j-1](k=y)$

所以 $f[i][j](k=y) \leqslant f[i][j](k=x)$

也就是 $x < y$ 一定不会更优

所以 $s[i][j] \geqslant s[i][j-1]$ 得证。

同理可证 $s[i][j] \leqslant s[i+1][j]$。

那么现在我们计算 $f[i][j]$ 的时候，不用在 $[i, j)$ 中枚举，而是在 $[s[i][j-1], s[i+1][j]]$ 里枚举就可以了。

对于所有的 $i$，$j$，可以证明区间 $[s[i][j-1], s[i+1][j]]$ 长度之和是 $O(n^2)$ 级别的。

# 合并果子

```
int f[N][N], s[N][N], sum[N];

for(int i = 1;i <= n;i ++){
        scanf("%d", &sum[i]);
        sum[i] += sum[i - 1];
}
for (int i = 1;i <= n;i ++){
        f[i][i] = 0;
        s[i][i] = i;
}
```

# 合并果子

```
for (int i = n; i; i --){
        for(int j = i + 1;j <= n;j ++){
                int Min = 1e9, p;
                for(int k = s[i][j-1];k <= s[i+1][j];k ++){
                        if (Min > f[i][k] + f[k+1][j]){
                                Min = f[i][k] + f[k+1][j];
                                p = k;
                        }
                }
                f[i][j] = Min + sum[j] - sum[i - 1];
                s[i][j] = p;
        }
}
```

# 实现技巧

比赛时，我们只要先写出暴力的 DP 然后打出 cost、dp、和决策数组，验证 cost、dp函数是否满足四边形不等式，以及决策是否具有单调性。

大胆猜想，小心求证！

下节课再见