

C++ 算法

动态规划

dynamic programming

一维DP

当动态规划问题的描述可以只用一个数字代表时，称为一维DP。这个数字描述的就是状态。

什么是状态？

一维DP 也称作 线性DP

例题：魔鬼的步伐

计数问题

魔鬼共有 n 级楼梯要走，魔鬼有他的步伐，每一步他只可以向上走 a 级楼梯或者 b 级楼梯，请问共有多少种不同的走法可以正好走完 n 级台阶。输入正整数 n ， a 和 b 。 $n, a, b \leq 50$ 。 a 不等于 b 。

输入样例：

4 3 5

输入样例：

10 2 5

输出样例：

0

输出样例：

2

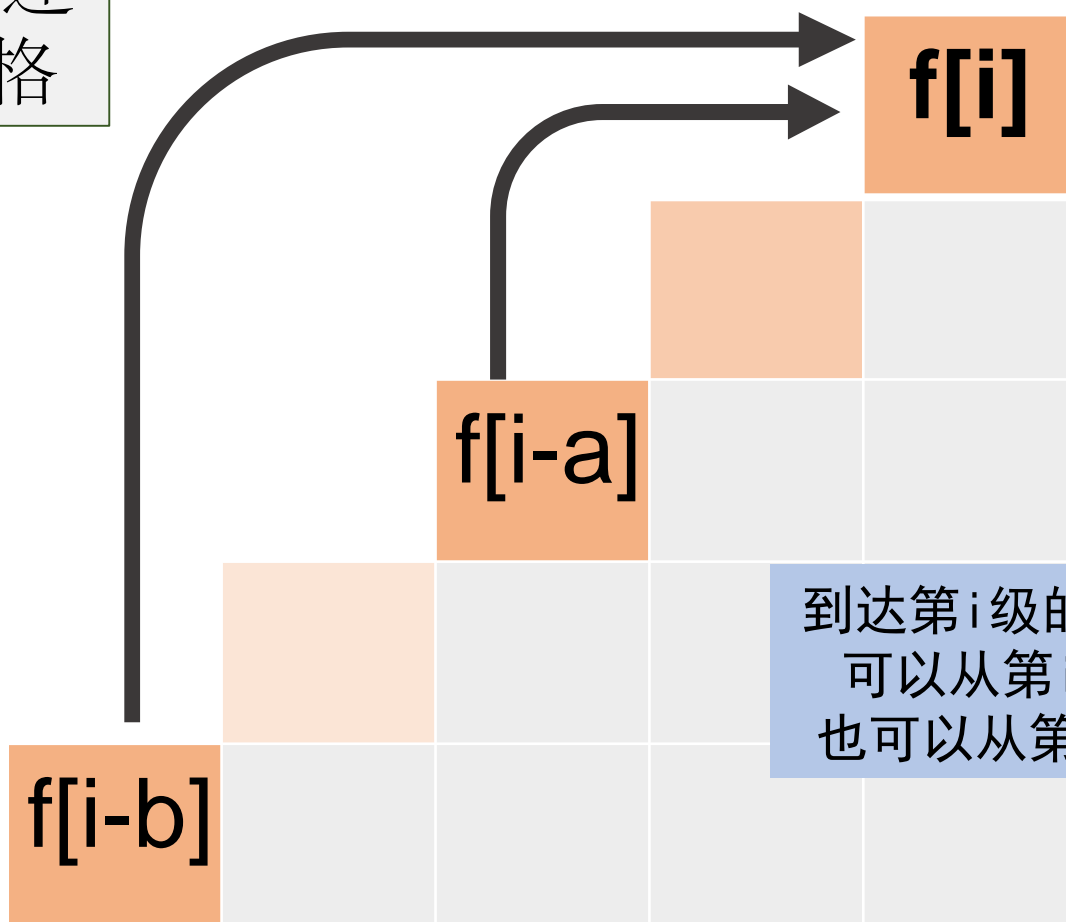
一步一步
从低到高走

一步一步
从低到高
递推求解

魔鬼的步伐

$f[i]$ 代表走到第*i*级共有多少种方法

状态*i*描述
在哪一格



到达第*i*级的方式有两种：
可以从第*i-a*级走*a*级来
也可以从第*i-b*级走*b*级来

魔鬼的步伐

$f[i]$ 代表走到第*i*级共有多少种方法

状态*i*描述
在哪一格

当*i*为0时

$$f[0] = 1$$

初始条件

当*i*大于0时

$$f[i] = f[i - a] | i \geq a \\ + f[i - b] | i \geq b$$

若 $i \geq a$, 可走a级来

若 $i \geq b$, 可走b级来

状态转移
方程

请填空

```
1  #include<iostream>
2  using namespace std;
3  const int N=1009;
4  const int INF=2000;
5  int n,a,b,f[N];
6  int main(){
7      cin>>n>>a>>b;
8      for(int i=0;i<=n;i++)f[i]=INF;
9      f[0]=;
10     for(int i=1;i<=n;i++){
11         if(i>=a)f[i]=min(f[i],f[i-a]+1);
12         if(i>=b)f[i]=min(,);
13     }
14     if(f[n]==)cout<<-1<<endl;
15     else cout<<f[><<endl;
16     return 0;
17 }
```

DP的三类问题

计数问题

魔鬼共有 n 级楼梯要走，魔鬼有他的步伐，每一步他只可以向上走 a 级楼梯或者 b 级楼梯，请问**共有多少种不同的走法正好走完 n 级台阶？**

可行性问题

魔鬼共有 n 级楼梯要走，魔鬼有他的步伐，每一步他只可以向上走 a 级楼梯或者 b 级楼梯，请问**能否走到第 n 级台阶？**

最优化问题

魔鬼共有 n 级楼梯要走，魔鬼有他的步伐，每一步他只可以向上走 a 级楼梯或者 b 级楼梯，请问**走到第 n 级台阶至少要几步？走不到时输出-1**

例题：最大连续子序列和

输入 n ，再依次输入 n 个整数组成的数组，第 i 个数为 $x[i]$ ，求数组中最大连续子序列和（至少包含一个数字）

输入样例

5
1 3 -2 4 -5

输入样例

6
-2 11 -4 13 -5 -2

输入样例

3
-3 -2 -1

输出样例

6

输出样例

20

输出样例

-1

思考：如何设计状态？
如何设计 $f[i]$ 代表什么含义？
如何建立 $f[i]$ 和 $f[i-1]$ 的联系？

例题：最大连续子序列和

$f[i]$ 代表以第 i 个数结尾的连续和最大值

状态 i 描述以
哪个数结尾



计算 $f[i]$ 时考虑两种可能：

1. 延用 $f[i-1]$ 结果
2. 舍弃 $f[i-1]$ 结果

最大连续子序列和

$f[i]$ 代表以第*i*个数结尾的连续和最大值

状态*i*描述以
哪个数结尾

当*i*为0时

$$f[0] = x[0]$$

初始条件

当*i*≥1时

$$f[i] = \max(f[i-1], 0) + x[i]$$

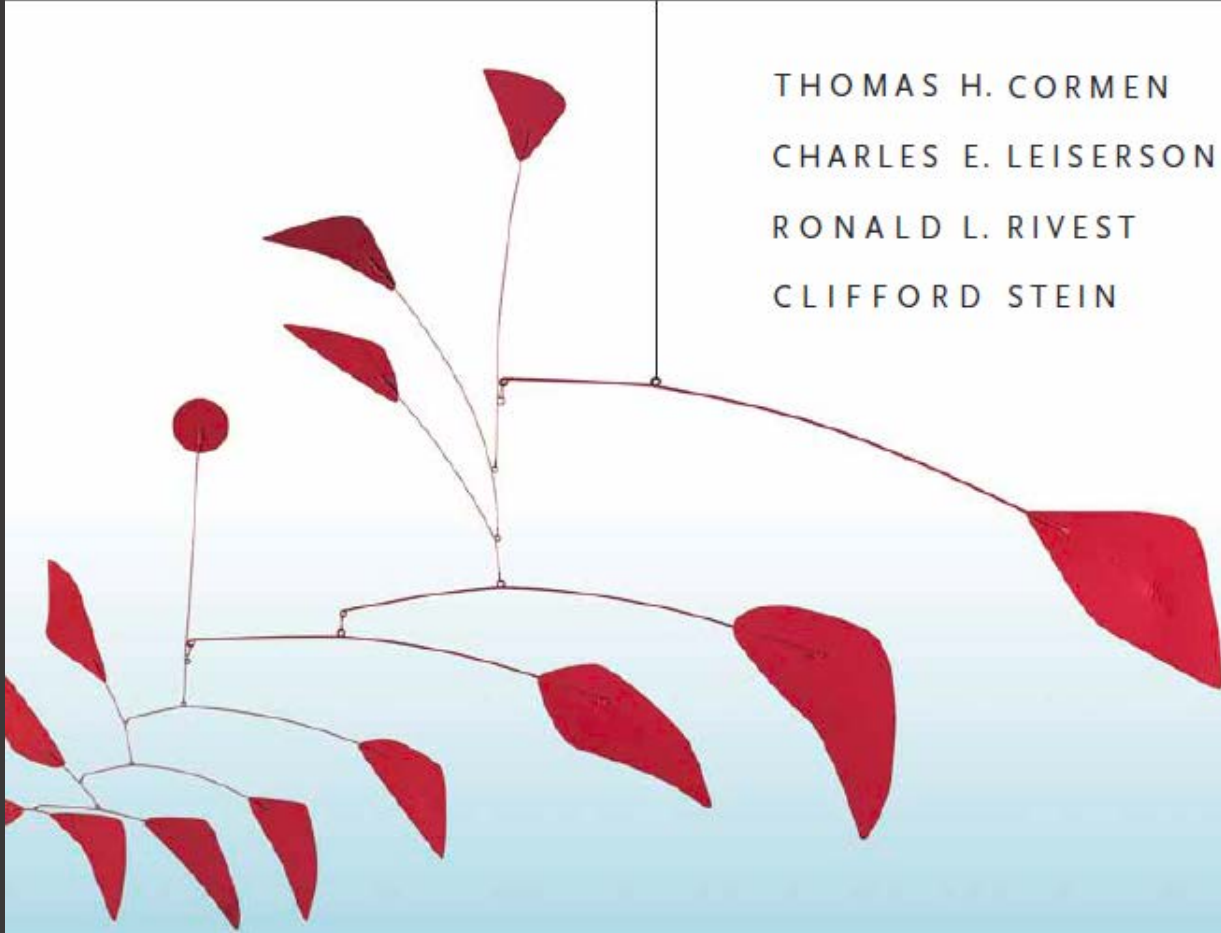
状态转移
方程

计算 $f[i]$ 时参考 $f[i-1]$ 结果
状态 $[i]$ 和状态 $[i-1]$ 之间“转移”

$$ans = \max_i \{f[i]\}$$

最终答案

枚举所有可能的*i*, 找 $f[i]$ 最大值



THOMAS H. CORMEN
CHARLES E. LEISERSON
RONALD L. RIVEST
CLIFFORD STEIN

INTRODUCTION TO

ALGORITHMS

THIRD EDITION

例题：钢条切割1

The rod-cutting problem is the following. Given a rod of length n inches and a table of prices p_i for $i=1,2,\dots,n$. Determine the maximum revenue obtainable by cutting up the rod and selling the pieces. Note that if the price p_n for a rod of length n is large enough, an optimal solution may require no cutting at all. $n \leq 100$

输入样例

8

1 5 8 9 10 17 17 20

输入第一行为 n 代表钢条长度。输入第二行的 n 个数字为 p_i ,代表长度为 i 的钢条不切割时能卖多少钱

输出样例

22

思考：如何设计状态？

如何设计 $f[i]$ 代表什么含义？

如何建立 $f[i]$ 和 $f[i-1], f[i-2], \dots$ 的联系？

选自《算法导论》
书名简称CLRS

钢条切割1

$f[i]$ 代表长度为*i*的钢条最多能卖多少钱

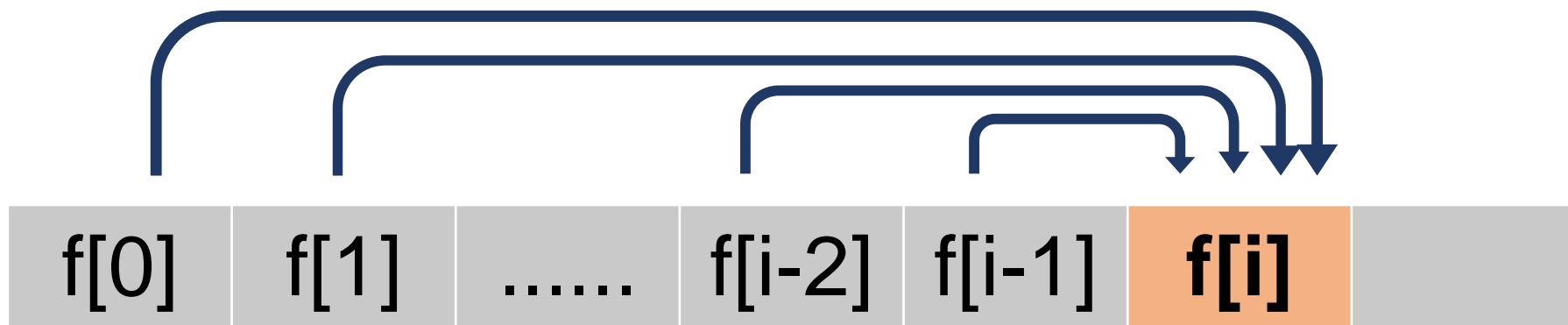
状态*i*描述
钢条长度

i	p[i]	f[i]	切割方案
1	1	1	无切割
2	5	5	无切割
3	8	8	无切割
4	9	10	$4 = 2 + 2$
5	10	13	$5 = 2 + 3$
6	17	17	无切割
7	17	18	$7 = 1 + 6$ 或 $7 = 2 + 2 + 3$
8	20	22	$8 = 2 + 6$

钢条切割1

状态*i*描述
钢条长度

$f[i]$ 代表长度为*i*的钢条最多能卖多少钱



计算 $f[i]$ 时参考
 $f[0], f[1], \dots, f[i-1]$ 的结果

钢条切割1

$f[i]$ 代表长度为*i*的钢条最多能卖多少钱

状态*i*描述
钢条长度

当*i*为0时

$$f[0] = 0$$

初始条件

当*i*≥1时

$$f[i] = \max_{1 \leq j \leq i} \{f[i-j] + p[j]\}$$

状态转移
方程

计算 $f[i]$ 时参考
 $f[0], f[1], \dots, f[i-1]$ 的结果

$$ans = f[n]$$

最终答案

复杂度 $O(N^2)$

例题：最长下降子序列

The **longest decreasing subsequence (LDS)** problem is to find a subsequence of a given sequence in which the subsequence's elements are in sorted order, highest to lowest, and in which the subsequence is as long as possible.

输入样例

8

7 8 4 6 8 9 3 2

输入第一行为 n 代表序列长度。

输入第二行为序列的 n 个数字： $x[0], x[1], \dots, x[n-1]$

输出下降子序列最长的长度。

输出样例

4

思考：如何设计状态？

如何设计 $f[i]$ 代表什么含义？

如何建立 $f[i]$ 和 $f[i-1], f[i-2], \dots$ 的联系？

各种子序列(标记红色)

7 8 4 6 8 9 3 2

上升子序列

7 8 4 6 8 9 3 2

不升子序列

7 8 4 6 8 9 3 2

下降子序列

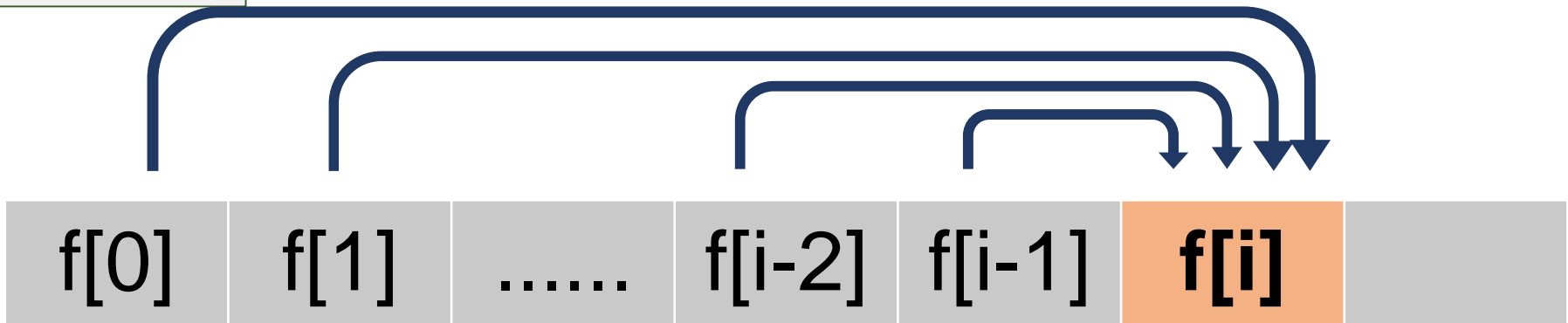
7 8 4 6 8 9 3 2

下降子序列

最长下降子序列

$f[i]$ 代表以 $x[i]$ 结尾的下降子序列最长能有多长

状态*i*描述
子序列结
尾的位置



计算 $f[i]$ 时参考
 $f[0], f[1], \dots, f[i-1]$ 的结果

最长下降子序列

$f[i]$ 代表以 $x[i]$ 结尾的下降子序列最长能有多长

状态 i 描述
子序列结
尾的位置

当 i 为0时

$$f[0] = 1$$

初始条件

当 $i \geq 1$ 时

$$f[i] = \max_{0 \leq j \leq i-1} \{f[j] | x[j] > x[i]\} + 1$$

状态转移
方程

计算 $f[i]$ 时参考
 $f[0], f[1], \dots, f[i-1]$ 的结果

$$ans = \max_i \{f[i]\}$$

最终答案

复杂度 $O(N^2)$, 能否更快?

记录决策：钢条切割2

输入第一行为 n 代表钢条长度。输入第二行的 n 个数字为 p_i ,代表长度为 i 的钢条不切割时能卖多少钱。

输出长度为 n 的钢条最多能卖多少钱，以及一种切割方案。

输入样例

8

1 5 8 9 10 17 17 20

输出样例

22

8=2+6

思考：

计算 $f[i]$ 时，如何记录决策？

记录决策

$f[i]$ 代表长度为 i 的钢条最多能卖多少钱

$fd[i]$ 记录计算 $f[i]$ 时的决策：选择切割多长

```
8      f[0]=0; fd[0]=0;
9      for(int i=1; i<=n; i++)
10         for(int j=1; j<=i; j++)
11             if(f[i-j]+p[j]>f[i]){
12                 f[i]=f[i-j]+p[j];
13                 fd[i]=j;
14             }
```

```
16      cout<<n<<"="<<fd[n];
17      while(1){
18          n-=fd[n];
19          if(n==0) break;
20          cout<<"+"<<fd[n];
21      }
```

一维DP总结

用变量 i 描述状态，定义 $f[i]$ 的含义

状态转移方程：联系 $f[i]$ 与 $f[i-1], f[i-2], \dots, f[0]$

初始化 f 数组

计算 $f[i]$ 时可记录决策



子问题自相似

DP解题步骤

定义
状态

写明一维数组元素 $f[i]$ 的含义

手算
填表

手算样例对应的填表结果
获取灵感，总结规律

初始化
+
状态转移

数组 $f[i]$ 的初始化和状态转移

一维DP的查错

静态 查错

写明一维数组元素 $f[i]$ 的含义

检查数组 $f[i]$ 的初始化和状态转移

检查数组下标准确性，包括越界可能

打印 日志

打印一维数组中每个元素 $f[i]$

查看每一格是否填写正确

由错误的那个 $f[i]$ 元素定位程序错误

作业要求

作业要求如下,老师会逐个程序检查:
程序开头需要先用注释写清楚数组含义, 以及手算样例对应的填表结果
例如对第一题:

```
/*
```

```
b[i]代表输入序列第i个数
```

```
f[i]代表以i号数字结尾的最长上升子序列的长度
```

```
i = 0,1,2,3,4
```

```
b[i]= 2,1,1,2,3
```

```
f[i] = 1,1,1,2,3
```

```
*/
```