1.1 (1) 结束时间早的优先

$w_1=1$ $w_2=5$

贪心解：w_1　　最优解：w_2

(2) 权重高的区间优先

$w_1=4$ $w_2=4$ $w_3=5$

贪心解：w_3　　最优解：w_1, w_2

(3) 冲突少的区间优先

$w_1=2$ $w_2=3$ $w_3=10$ $w_4=12$

贪心解：w_1, w_2　　最优解：w_4

1.2

	A_1	A_2	A_3	A_4
(1)	3×6	6×9	9×5	5×4
(2)	2×5	5×10	10×7	7×8

(1) opt(i,j) 表

	1	2	3	4	
	0	162	297	357	1
		0	270	390	2
			0	180	3
				0	4

$opt(1,2) = 3\times6\times9 = 162$
$opt(2,3) = 6\times9\times5 = 270$
$opt(3,4) = 9\times5\times4 = 180$
$opt(1,3) = opt(1,2) + 3\times9\times5 = 297$
$opt(2,4) = opt(2,3) + 6\times5\times4 = 390$
$opt(1,4) = opt(1,3) + opt(4,4) + 3\times5\times4 = 357$

(2) 计算顺序为 $((A_1A_2)A_3)A_4$

(2) opt(i,j)表

	1	2	3	4
1	0	100	240	
2		0	350	630
3			0	560
4				0

$opt(1,2) = 2\times5\times10 = 100$

$opt(2,3) = 5\times10\times7 = 350$

$opt(3,4) = 10\times7\times8 = 560$

$opt(1,3) = opt(1,2) + 2\times10\times7 = 240$

$opt(2,4) = opt(2,3) + 5\times7\times8 = 630$

$opt(1,4) = opt(1,3) + 2\times7\times8 = 352$

∴ 计算顺序为 $((A_1A_2)A_3)A_4$

1.3 (1) 用一个二维数组opt[v][k]记录方问题的解

v表示当前顶点，k表示最多经过k条边（k=0, 1, 2, ..., n-1）

初始化：opt[v][0] = ∞ , opt[t][0] = 0

计算顺序：外层循环j从1到n-1，内层循环i遍历每个点，计算opt[i][j]

(2) $opt(v,k) = \min(opt(v,k-1), \min_{(w,v)\in E}(opt(w,k-1) + C(w,v)))$

边界条件 $opt(v,k) = \infty$ $opt(s,k) = 0$

输出结果：opt(v, n-1)，计算过程同(1)

1.4 从后向前处理

定义方问题 $f_i(X)$ 表示 KNAP(1, i, X) 的最优解

$$f_i(X) = \begin{cases} 0, & i<0 \\ -\infty, & X<0 \\ \max(f_{i-1}(X), f_{i-1}(X-W_i)+P_i) \end{cases}$$

最终返回 $f_n(m)$ 即可

1.5 设爬i阶楼梯的方法数为 $f(i)$ $(i\geq 2)$

则 $f(i) = f(i-1) + f(i-2)$

初始条件：$f(0)=1$，$f(1)=1$

伪代码为：

```
climbStairs(n):
if n==0 or n==1:
    return 1
return climbStairs(n-1)+climbStairs(n-2)
```

时间复杂度为 ~~O(n)~~ $O(2^n)$

如果使用记忆化的递归或用循环代替递归，时间复杂度可降为 $O(n)$

1.6 (1) 每根柱子上的接雨量取决于左侧和右侧最高柱子的较小值

$leftMax[i] = max(leftMax[i-1], h[i-1])$

$rightMax[i] = max(rightMax[i+1], h[i+1])$

$leftMax[1] = 0$

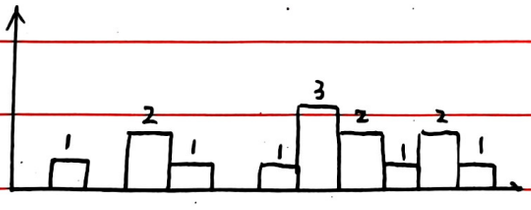
$rightMax[n] =$

其中，$leftMax[2] = h[1]$　　$rightMax[n-1] = h[n]$

$water[i] = max(min(leftMax[i], rightMax[i]) - h[i], 0)$

$total = \sum_{i=1}^{n} water[i]$

(2)



$h[] = [0,1,0,2,1,0,1,3,2,1,2,1]$

$leftMax = [0,0,1,1,2,2,2,2,3,3,3,3]$

$rightMax = [3,3,3,3,3,3,3,2,2,2,1,0]$

$\therefore water = [0,0,1,0,1,2,1,0,0,1,0,0]$

$\therefore total = 6$

~~1.7 设 $f_i(L)$ 表示将L的钢条最大长度为i时切割方案采用的最大收益~~

$$\therefore f_i(L)=\begin{cases} \max\left(f_{i-1}(L),\ \max\limits_{c\in N^+} f_{i-1}(L-ci)+cp_i\right) \\ 0, & i<0 \\ -\infty, & L<0 \end{cases}$$

1.7 设 $dp(L)$ 为长度L的钢条的最大收益

则 $dp(L) = \max\limits_{1\le i\le L} (p_i + dp(L-i))$

当 $L\le 0$，$dp(L)=0$

计算顺序：自底向上，从1到L依次求解

伪代码：

```
maxValue(L, p)
Init dp[0 ... L] = 0
for j = 1 to L
    maxp = 0
    for i = 1 to j
        maxp = max(maxp, p[i] + dp[j-i])
    dp[j] = maxp
return dp[L]
```