例0 找零钱

- 顾客来买东西，要找给他85元，通常怎么给他找零？

例0 找零钱

- 顾客来买东西，要找给他85元，通常怎么给他找零？
- 通常——隐含着使用最少的纸币张数

例0 找零钱

- 顾客来买东西，要找给他85元，通常怎么给他找零？
- 通常——隐含着使用最少的纸币张数
- 贪心是人类自带的能力

例0' 找零钱

- 顾客来买东西，要找给他85元，通常怎么给他找零？
- 假如当前流通的纸币面额是50元、40元、10元、5元呢？

例0' 找零钱

- 顾客来买东西，要找给他85元，通常怎么给他找零？
- 假如当前流通的纸币面额是50元、40元、10元、5元呢？
- 在这种情况下，优先使用大额纸币是不正确的贪心选择。

例0'' 找零钱

- 有**1**元、**2**元、**5**元、**10**元的纸币分别为**a, b, c, d**张，要用这些纸币凑出**m**元，至少要用多少张纸币**?**

- 样例输入：
- **100**
- **10 20 8 5**
- 样例输出：
- **18**

例0'' 找零钱

- 这个题怎么做？

例0'' 找零钱

- 递归？
- 递推？

- 递归：

例0'' 找零钱

- 递推（动态规划）：
- 要凑出m元，必须先凑出m-1、m-2、m-5、m-10元，我们用dp[i]表示凑出i元的最少纸币数;
- 有 dp[i]=min(dp[i-1], dp[i-2], dp[i-5], dp[i-10]) + 1;
- 容易知道dp[1]=dp[2]=dp[5]=dp[10]=1;
- 根据以上递推方程和初始化信息，可以容易推出dp[1]……dp[m]的所有值。

例0'' 找零钱

- 似乎有些不对?平时我们找零钱有这么复杂吗?

- 当一个问题具有最优子结构性质时，我们会想到用动态规划法去解它。但有时会有更简单有效的算法。用贪心算法更简单，更直接且解题效率更高。这利用了问题本身的一些特性。例如，上述通常找零钱的算法便是利用了硬币面值的特殊性。

再举个例子：方格取数问题

- 在一个N×M的方格阵中，每一格子赋予一个数（即为权）。规定每次移动时只能向上或向右。现试找出一条路径，使其从左下角至右上角所经过格子的权之和最大。

30	4	6
1	2	100

- 本题用贪心策略不能得到最优解。
- 若按贪心策略求解，所得路径为：**1,30,4,6**；
- 若按动态规划法求解，所得路径为：**1,2,100,6**。

30	4	6
1	2	100

- 贪心算法总是作出在当前看来是最好的选择。也就是说贪心算法并不从整体最优上加以考虑，它所作出的选择只是在某种意义上的局部最优选择。

- 虽然贪心算法不是对所有问题都能得到整体最优解，但对范围相当广的许多问题它能产生整体最优解。如图的单源最短路径问题，最小生成树问题等。在一些情况下，即使贪心算法不能得到整体最优解，但其最终结果却是最优解的很好的近似解。

贪心法

sdsy _ n

- 有人说贪心算法是最简单的算法，原因很简单：
- 你我其实都很贪，根本不用学就知道怎么贪。

- 有人说贪心算法是最复杂的算法，原因也很简单：
- 这世上会贪的人太多了，哪轮到你我的份？

#1284. 排队打水

#624. [NOIP2002提高组] 均分纸牌

#1511. [NOI1994] 删数问题

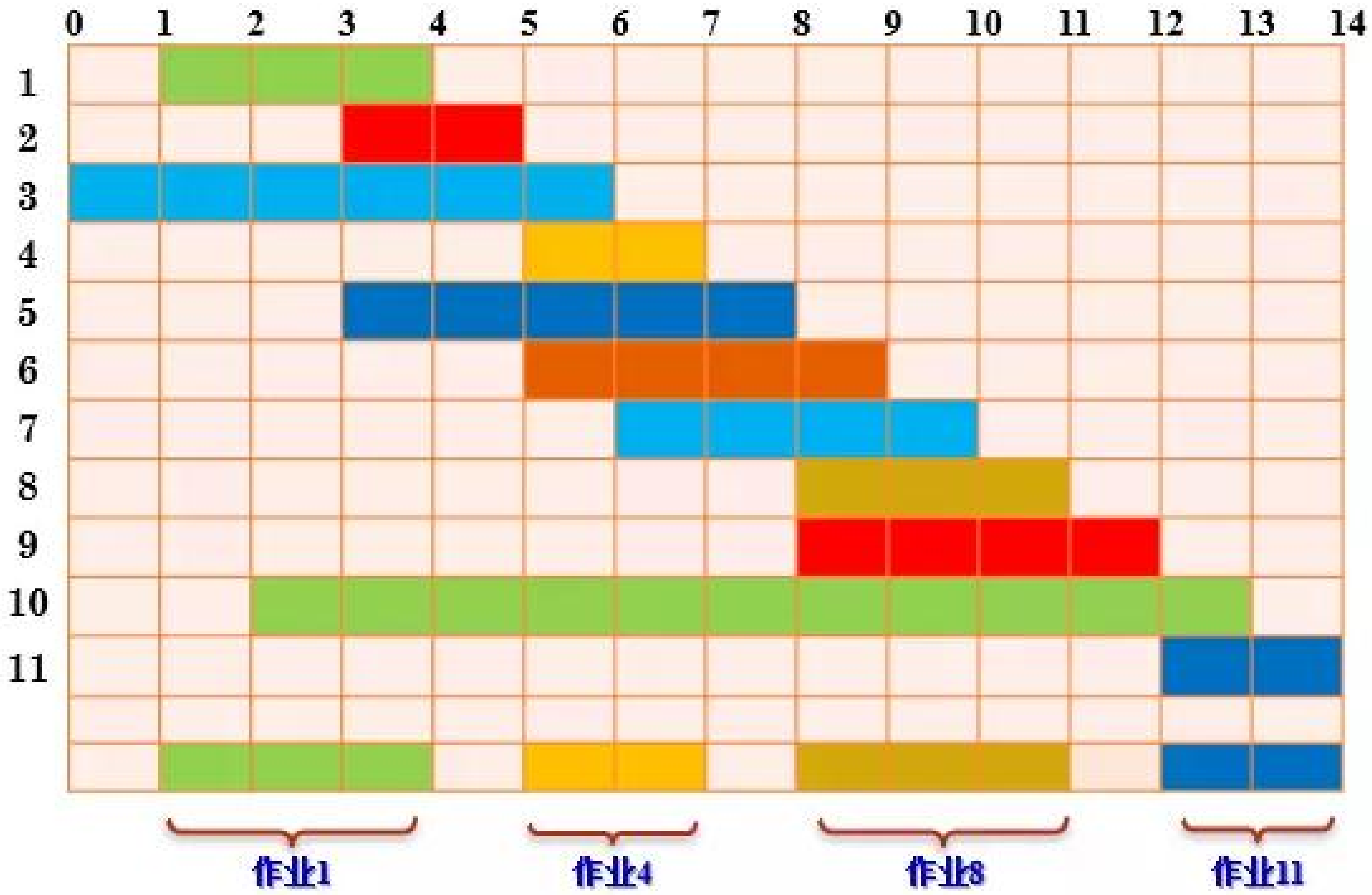
#454. [NOIP1998提高组] 拼数

#489. [NOIP1999提高组] 导弹拦截

#1501. 线段覆盖 1

- 样例：

i	1	2	3	4	5	6	7	8	9	10	11
s[i]	1	3	0	5	3	5	6	8	8	2	12
f[i]	4	5	6	7	8	9	10	11	12	13	14



活动安排问题的几何意义

贪心策略及证明

- 经典贪心问题：最多不相交区间数问题

三类经典的区间贪心问题

- **1、**最多不相交区间数问题**—#1501.** 线段覆盖 **1**
- **2、**区间完全覆盖问题**—#1504.** 线段覆盖
- **3、**区间最小点覆盖问题**—#1506.** 区间最小点覆盖

2、区间完全覆盖问题

- 问题描述：给定一个区间[s,t]，再给出n条线段的起点和终点[ai,bi]（注意这里是闭区间），求最少使用多少条线段可以将整个区间完全覆盖。

- 样例：给出区间[1,8]，可选的线段有[2,6]，[1,4]，[3,6]，[3,7]，[6,8]，[2,4]，[3,5]。

贪心策略

- **1**、对于当前区间**[a,b]**来说，选择的下一个区间的左端点值**a2**一定不会大于**b**，否则就不能完成“覆盖”这一操作。
- **2**、对于当前区间**[a,b]**来说，如果有多个区间都满足条件**1**，那么一定选择右端点最大的区间，否则就不能满足“最少”这一目的。

分析

- 把各线段按照左端点a从小到大排序。如果第一条线段的起点不是s，无解，否则第一条线段是起点在s的最长线段，即右端点b最大的线段。选择[ai,bi]后，新的起点设置成bi。依次向后，直至覆盖整个线段或无解。

样例求解过程：

- 1、将每一条线段按左端点递增顺序排列，如果左端点相同，按右端点递增顺序排列，排完序后为[1,4]，[2,4]，[2,6]，[3,5]，[3,6]，[3,7]，[6,8]；
- 2、设置一个变量表示已覆盖到的区间右端点，在剩下的线段中找出所有左端点小于等于当前已覆盖到的区间右端点的线段，选择右端点最大并且大于当前已覆盖到的区间右端点，重复以上操作直至覆盖整个区间；
- 3、模拟过程：假设第一次加入[1,4]，那么下一次能够选择的线段有[2,6]，[3,5]，[3,6]，[3,7]，由于3小于4且7最大，所以下一次选择[3,7]进行覆盖，最后一次只能选择[6,8]，这个时候刚好覆盖整个区间-->break；即所选3条线段就能覆盖整个区间；

对应习题：

- **NYOJ #12 喷水装置**

3、区间最小点覆盖问题

- 问题描述：
- 数轴上有n个闭区间 [ai,bi]，要求选取尽量少的点，使得每个区间内都至少有一个点（不同区间内含的点可以是同一个）。

求解过程：

- 1、按左端点递增顺序排序，如果左端点相同，按右端点递增顺序排序（这种比较好理解，当然也可以先按右端点递增顺序排序）。
- 2、①从第一个区间右端点开始贪心往后找，如果下一个区间的左端点大于当前已选区间的右端点，说明要新开一个点，计数器加1，同时更新右区间能覆盖的最远距离；②如果下一个区间右端点小于当前已选区间的右端点，说明共享的线段范围缩短了，那么就更新区间右端点为下一个区间右端点，重复以上操作，直至筛选完所有区间。

课后习题

- **NYOJ #891 找点**
- **poj 1328 Radar Installation**

例 截铁丝

- 现有长为144cm的铁丝，要截成n小段（n>2），每段的长度都是整数，且不小于1cm，如果其中任意三小段都不能拼成三角形，则n的最大值为？怎么截？

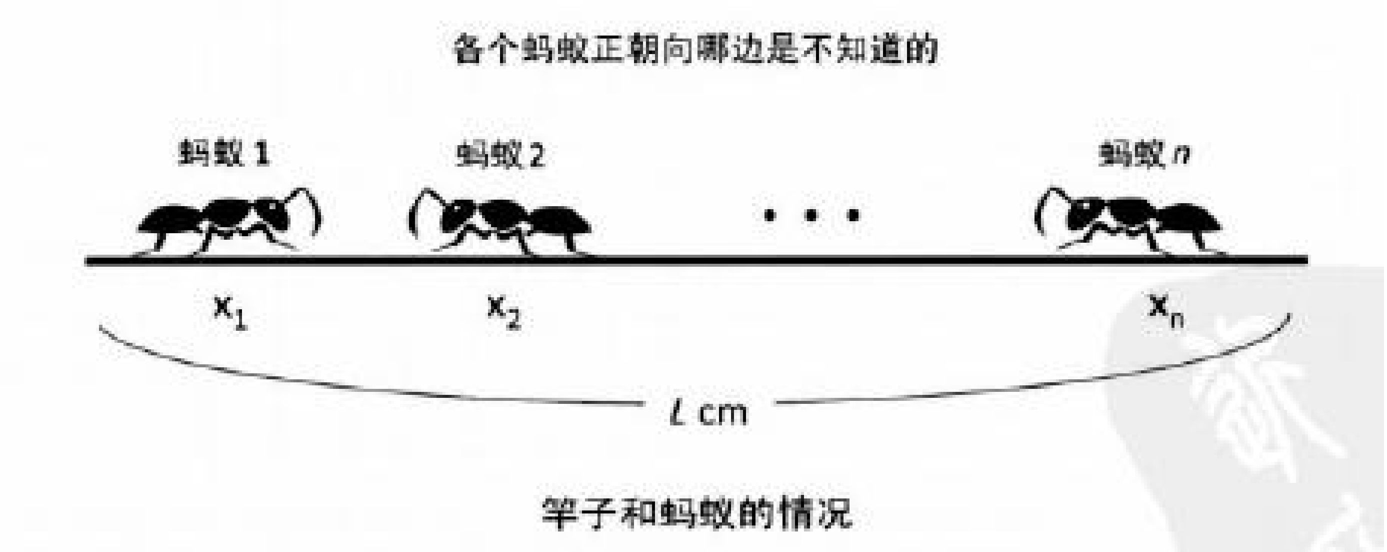
例 石子游戏

- 有一堆小石子共**100**颗。甲、乙两人轮流取，每人每次取的石子不能超过**2**颗，且不能不取。规定能取到最后一颗石子的人为胜者。若甲先取，他有无必胜把握？若有，他该怎么取？若没有，说明理由。

例 Ants（POJ1852）

- 有N只蚂蚁（蚂蚁的长度忽略不计）在长度为 L cm的水平杆上行走，每只蚂蚁都具有1 cm / s的恒定速度。当一只行走的蚂蚁到达杆的一端，它立即掉落。当两只蚂蚁相遇时，他们同时掉头（掉头时间忽略不计）朝着相反的方向行走。我们知道每只蚂蚁在杆子上的初始位置（距离水平杆最左端的距离），不幸的是，我们不知道蚂蚁初始行走的方向。你的任务是计算所有蚂蚁脱离杆子所需的最短和最长的可能时间。

- 例如，当N=3, L=10, 3只蚂蚁的初始位置为2, 6, 7时，最短和最长可能时间分别是4和8。
- 则当 N=7，L=214, 7只蚂蚁的初始坐标为11, 12, 7, 13, 176, 23, 191时，最短和最长可能时间分别是__________和__________。



例 纪念品分组 NOIP普及组2007

- 问题描述
- 元旦快到了，校学生会让乐乐负责新年晚会的纪念品发放工作。为使得参加晚会的同学所获得的纪念品价值相对均衡，他要把购来的纪念品根据价格进行分组，但每组最多只能包括两件纪念品，并且每组纪念品的价格之和不能超过一个给定的整数。为了保证在尽量短的时 间内发完所有纪念品，乐乐希望分组的数目最少。
- 你的任务是写一个程序，找出所有分组方案中分组数最少的一种，输出最少的分组数目。

- 输入格式
- 输入包含n+2行：
- 第1行包括一个整数w，为每组纪念品价格之和的上限。
- 第2行为一个整数n，表示购来的纪念品的总件数。
- 第3~n+2行每行包含一个正整数pi (5 <= pi <= w)，表示所对应纪念品的价格。
- 输出格式
- 输出仅一行，包含一个整数，即最少的分组数目。

- 样例输入
- **100**
- **9**
- **90**
- **20**
- **20**
- **30**
- **50**
- **60**
- **70**
- **80**
- **90**

- 样例输出
- 6

- 数据规模和约定
- 50%的数据满足：1 <= n <= 15
- 100%的数据满足：1 <= n <= 30000, 80 <= w <= 200

分析

#283. [NOIP 2004 提高组] 合并果子 ++

例 石子合并

有N堆石子(N≤100)排成一排。现要将石子合并成一堆.规定每次只能选相邻的两堆合并成一堆新的石子,并将新的一堆的石子数,记为该次合并的得分.

选择一种合并石子的方案,使得做N-1次合并,得分的总和最少。

输入数据:

第一行为石子堆数N;

第二行为每堆石子数。

输出数据：

合并石子后得到的最小得分。

- 样例输入:

4

1 3 5 2

- 样例输出

22

分析

- 贪心？
- 如何证明正确或不正确？

- 反例：
- **7 4 4 7**

- 石子合并是经典的区间型动态规划问题。
- 课下自学。

- 注重代码落实！

The end.

Thanks.