

例谈贪心

安徽师范大学附属中学 叶国平

引例-背包问题

- 有一个背包，背包容量是 $M=150\text{kg}$ 。有7个物品，物品不可以分割成任意大小。要求尽可能让装入背包中的物品总价值最大，但不能超过总容量。

重量	35kg	30kg	60kg	50kg	40kg	10kg	25kg
价值	10\$	40\$	30\$	50\$	35\$	40\$	30\$

引例-背包问题

- 策略一：每次挑选价值最大的物品装入背包，得到的结果是否最优？(165\$)
- 策略二：每次挑选所占重量最小的物品装入是否能得到最优解？(155\$)
- 策略三：每次选取单位重量价值最大的物品装入是否能得到最优解？(170\$)

引例-背包问题

- 有一个背包，背包容量是 $M=150\text{kg}$ 。有7个物品，物品不可以分割成任意大小。要求尽可能让装入背包中的物品总价值最大，但不能超过总容量。

重量	15kg	30kg	60kg	50kg	40kg	10kg	25kg
价值	10\$	40\$	40\$	50\$	35\$	40\$	30\$

引例-背包问题

- 策略一：每次挑选价值最大的物品装入背包，得到的结果是否最优？(170\$)
- 策略二：每次挑选所占重量最小的物品装入是否能得到最优解？(155\$)
- 策略三：每次选取单位重量价值最大的物品装入是否能得到最优解？(170\$)
- 但实际上采用动态规划来计算，我们能得到的最优解为175\$，不选25kg和60kg的两个物品。

什么是贪心算法

- 顾名思义，贪心算法总是作出在当前看来是最好的选择。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都能产生整体最优解或是问题的次优解。因此有很好研究它的必要。
- 贪心是一种解题策略，也是一种解题思想。

基本要素-贪心策略

- 贪心选择是指所求问题的整体最优解可以通过一系列局部最优的选择，即贪心选择来达到。这是贪心算法可行的第一个基本要素。可通过局部的贪心选择来达到问题的全局最优解。运用贪心策略解题，一般来说需要一步步的进行多次的贪心选择。在经过一次贪心选择之后，原问题将变成一个相似的，但规模更小的问题，而后的每一步都是当前看似最佳的选择，且每一个选择都仅做一次。

基本要素-最优子结构

- 原问题的最优解包含子问题的最优解，即问题具有最优子结构的性质。在背包问题中，第一次选择单位质量最大的货物，它是第一个子问题的最优解，第二次选择剩下的货物中单位重量价值最大的货物，同样是第二个子问题的最优解，依次类推。但并不是所有具有最优子结构的问题都可以用贪心策略求解。因为贪心往往是盲目的，需要使用更理性的方法——动态规划。

基本要素-最优子结构

- 贪心算法和分治法及动态规划三者都具有子结构，都是将原问题归纳为更小规模的相似的子问题，并通过求解子问题，最后获得整体解，三者有何不同呢？只有搞清三者的区别，才能很好地运用它们来解决问题。
- 首先是它们的子问题（或称子结构）是不同的。分治法中各子结构是独立的，动态规划一般具有重叠最优子结构，除了必须满足具有最优子结构外，还要满足无后效性。贪心算法要求问题具有最优子结构。

基本要素-最优子结构

- 其次，在算法实现上看，分治法一旦递归地求出各子结构的解后，便可自下而上地将这些子结构解合并成问题的解。动态规划是所有子问题只计算一次并记录下来，以备后面的子问题使用，用空间换取时间，所以求当前的解要依赖前面子结构的解。一般采用自底向上的递推方式求解。贪心法则是从上而下，从问题的初始阶段开始每个阶段作一个贪心的选择，不断将问题转换为规模更小的子问题，并期望通过每一次的局部最优选择达到全局最优。

贪心算法：基本过程

- 建立数学模型来描述问题；
- 把求解的问题分成若干个子问题；
- 对每一子问题求解，得到子问题的局部最优解；
- 把子问题的解局部最优解合成原来解问题的一个解。
- 用好贪心算法关键是对一个问题能否运用贪心算法的判断和贪心标准的设计。我们主要还是根据贪心算法的两个性质，以及运用反证法来证明。更重要的是要通过编程实践多摸索、总结，从而掌握贪心算法。

例题：排队问题（贪心策略明显）

- 在一个医院B超室，有 n 个人要做不同身体部位的B超，已知每个人需要处理的时间为 $t[i]$ ($1 \leq i \leq n$)，请求出一种排列次序，使每个人排队等候时间总和最小。
- 输入第1行一个正整数 n ($n \leq 10000$)，第2行有 n 个不超过1000的正整数 t_i 。
- 输出要求： n 个人排队时间最小总和。
- 输入样例： 4 输出样例： 67
 5 10 8 7

分析

- n 个人时间从小到大排序，就是这 n 个人最佳排队方案。求部分和的和即为所求。
- 反证法证明：假设有最优解序列： $s_1, s_2 \dots s_n$, 如 s_1 不是最小的 T_{\min} ，不妨设 $s_k = T_{\min}$, 将 s_1 与 s_k 对调，显然，对 s_k 之后的人无影响，对 s_k 之前的人等待都减少了， $(s_1 - s_k) > 0$ ，从而新的序列比原最优序列好，这与假设矛盾，故 s_1 为最小时间，同理可证 $s_2 \dots s_n$ 依次最小。

例题：神牛果（贪心策略明显）

- 在某次膜拜大会上， n 个神牛被要求集体膜拜。这些神牛被奖励每人吃一些神牛果。但是，每个神牛的肚量不一样。为了不显得某些人吃得太多，决定两人一组，使得吃得最多的那组吃得尽量少。
- 输入：第一行一个偶数 n ($n \leq 10000$)。第二行有 n 个正整数，为给定的一系列数字，表示每个神牛能吃多少神牛果。
- 输出：一个正整数，吃的最多的一组神牛吃的个数的最小值。

分析

- 将每个神牛能吃神牛果数按从小到大的顺序排序，在每一轮中选择最多的与最少的进行组合。
- 也可用反证法进行证明。

例题：修理牛棚（贪心策略明显）

- John的牛棚一个紧挨着另一个排成一行，有些牛棚里有牛，有些没有，所有的牛棚有相同的宽度。自门遗失以后，农民John必须尽快在牛棚前面竖立起新的木板。他的新木材供应商将会供应他任何他想要的长度，但是供应商只能提供有限数目的木板。农民John想将他购买的木板总长度减到最少。
- 给出能买到的木板最大的数目 M ，牛棚的总数 $S(1 \leq S \leq 200)$ ，牛棚里牛的总数 $C(1 \leq C \leq S)$ ，和牛所在的牛棚的编号 $X(1 \leq X \leq S)$ 。输出所需木板的最小总长度作为答案。

例题：修理牛棚（贪心策略明显）

➤ 输入：第1行三个数M，S和C(用空格分开)；第2到C+1行：每行包含一个整数，表示牛所占的牛棚的编号。

➤ 输出：一个整数表示所需木板的最小总长度。

➤ 输入样例：

4 50 18

3 4 6 8 14 15 16 17 21 25 26 27 30 31 40 41 42 43

➤ 输出样例：

25

例题：删数（贪心策略明显）

- 给定 n ($n \leq 100$) 位正整数 a ，去掉其中任意 $k \leq n$ 个数字后，剩下的数字按原次序排列组成一个新的正整数。对于给定的 n 位正整数 a 和正整数 k ，设计一个算法找出剩下数字组成的新数最小的删数方案。
- 输入：第1行是1个正整数 a ；第2行是正整数 k 。
- 输出：删去 k 个数字后得到的最小数。
- 输入样例： 178543 输出样例： 13
4

例题：奶酪工厂(贪心策略不明显)

- 接下来的 N ($1 \leq N \leq 10000$) 星期中，奶酪工厂在第 i 个星期要花 $C[i]$ 分来生产一个单位的奶酪。约克奶酪工厂拥有一个无限大的仓库，每个星期生产的多余的奶酪都会放在这里。而且每个星期存放一个单位的奶酪要花费 S 分。
- 工厂最近收到了客户 N 个星期的订单，第 i 个星期要向客户提供 $Y[i]$ 个单位的奶酪。当然这些奶酪可以在第 i 个星期时生产，也可以从仓库中拿取。采用怎样的生产策略约克奶酪工厂的花费最小呢？

例题：奶酪工厂(贪心策略不明显)

- 输入：第一行两个整数：N和S；接下来的N行中，第i行的两个数表示：C[i]和Y[i]。
- 输出：仅一行，即工厂生产的最小花费。
- 输入样例： 4 5 输出样例： 126900
88 200
89 400
97 300
91 500

例题：观光公交(贪心策略不明显)

- 风景迷人的小城Y市，拥有 n 个美丽的景点。Y市特意安排了一辆观光公交车，为游客提供更便捷的交通服务。观光公交车在第0分钟出现在1号景点，随后依次前往2、3、4..... n 号景点。从第 i 号景点开到第 $i+1$ 号景点需要 D_i 分钟。任意时刻，公交车只能往前开，或在景点处等待。设共有 m 个游客，每位游客需要乘车1次从一个景点到达另一个景点，第 i 位游客在 T_i 分钟来到景点 A_i ，希望乘车前往景点 B_i ($A_i < B_i$)。为了使所有乘客都能顺利到达目的地，公交车在每站都必须等待需要从该景点出发的所有乘客都上车后才能出发开往下一景点。假设乘客上下车不需要时间。

例题：观光公交(贪心策略不明显)

- 一个乘客的旅行时间，等于他到达目的地的时刻减去他来到出发地的时刻。因为只有一辆观光车，有时候还要停下来等其他乘客，乘客们纷纷抱怨旅行时间太长了。于是聪明的司机ZZ给公交车安装了 k 个氮气加速器，每使用一个加速器，可以使其中一个 D_i 减1。对于同一个 D_i 可以重复使用加速器，但是必须保证使用后 D_i 大于等于0。那么ZZ该如何安排使用加速器，才能使所有乘客的旅行时间总和最小？
- $1 \leq n \leq 1000$, $1 \leq m \leq 10000$, $0 \leq k \leq 100000$, $0 \leq D_i \leq 100$, $0 \leq T_i \leq 100000$

例题：观光公交(贪心策略不明显)

- 一个乘客的旅行时间，等于他到达目的地的时刻减去他来到出发地的时刻。因为只有一辆观光车，有时候还要停下来等其他乘客，乘客们纷纷抱怨旅行时间太长了。于是聪明的司机ZZ给公交车安装了 k 个氮气加速器，每使用一个加速器，可以使其中一个 D_i 减1。对于同一个 D_i 可以重复使用加速器，但是必须保证使用后 D_i 大于等于0。那么ZZ该如何安排使用加速器，才能使所有乘客的旅行时间总和最小？
- 共一行，包含一个整数，表示最小的总旅行时间。

例题：观光公交(贪心策略不明显)

➤ 输入样例：

3 3 2

1 4

0 1 3

1 1 2

5 2 3

输出样例： 10

贪心算法在数据结构中的应用

- 贪心算法在数据结构中的应用。求最小生成树的prim算法中，挑选的顶点是候选边中权值最小的边的一个端点，而加边法的kruskal算法中，更是首先将边的权值从小到大进行排序依次选取。
- 再看看求单源最短路径的算法。其基本算法思想是：设置顶点集合s并不断作贪心选择，来扩充这个集合，以最终求得最短路径。还有哈夫曼编码也用到贪心算法。

例题：堆积木

- 现在有 N 块积木，每块积木都有自重 W 和正常状态下的承重能力 F ，现在要把这 N 块积木垒在一起，但是有可能某块积木的负重超过了它在正常状态下的承重能力，那么这块积木就有被压坏的危险，请问应该如何堆这 N 块积木使得 N 块积木中最大的压力指数最小。
- 这里定义压力指数为该积木的负重与其在正常状态下的承重能力的差值。
- $1 \leq N \leq 50000$, $1 \leq W \leq 10000$, $1 \leq F \leq 10^9$ 。

例题：堆积木

➤ 输入：第一行为一个正整数 N ，表示有 N 块积木。第二行到第 $N+1$ 行，每行两个整数数，分别是第 i 个积木的 $W[i]$ 和 $F[i]$ 。

➤ 输出：输出共一行，表示最大压力指数的最小值。

➤ 输入样例： 3 输出样例： 2

10 3

2 5

3 3

例题：国王游戏

- 国王邀请 n 位大臣来玩一个有奖游戏。首先，他让每个大臣在左、右手上面分别写下一个整数，国王自己也在左、右手上各写一个整数。然后，让这 n 位大臣排成一排，国王始终站在队伍的最前面。排好队后，所有的大臣都会获得国王奖赏的若干金币，每位大臣获得的金币数分别是：排在该大臣前面的所有人的左手上的数的乘积除以他自己右手上的数，然后向下取整得到的结果。
- 国王想请你帮他重新安排一下队伍的顺序，使得获得奖赏最多的大臣，所获奖赏尽可能的少。

例题： 国王游戏

- 输入： 第一行包含一个整数 n ； 第二行包含两个整数 a 和 b ，分别表示国王左手和右手上的整数； 接下来 n 行，每行包含两个整数 a 和 b ，表示每个大臣左手和右手上的整数
- 输出： 包含一个整数，表示重新排列后的队伍中获奖赏最多的大臣所获得的金币数。

➤ 输入样例：

```
3
1 1
2 3
7 4
4 6
```

输出样例： 2

例题： 整数区间

- 我们定义一个整数区间 $[a, b]$ ：是一个从 a 开始至 b 结束的连续整数的集合。编一个程序，对给定的 n ($n \leq 1000$) 个区间，找出满足下述条件的所含元素个数最少的集合中元素的个数：对于所给定的每一个区间，都至少有两个不同的整数属于该集合。

例题：整数区间

- 输入：第一行一个正整数 n ，接下来有 n 行，每行给定一个区间的 a, b 值。
- 输出：一个正整数，满足条件的集合所包含的最少元素个数。
- 输入样例：

4
3 6
2 4
0 2
4 7

输出样例：

4

例题：雷达安装

- 假定海岸线是一条无限延伸的直线，陆地在这条海岸线的一边，大海在另一侧。海中有许多岛屿，每一个小岛我们可以认为是一个点。现在要在海岸线上安装雷达，雷达的覆盖范围是 d ，也就是说大海中一个小岛能被安装的雷达覆盖，那么它们之间的距离最大为 d 。
- 我们使用平面直角坐标系，定义海岸线是 x 轴，大海在 x 轴上方，陆地在此下方。给你海中每一个岛屿的坐标位置 (x,y) 和要安装的雷达所覆盖的范围 d ，你的任务是写一个程序计算出至少安装多少个雷达能将所有的岛屿覆盖。

例题：雷达安装

➤ 输入：第一行两个整数 n 和 d ，分别表示海中岛屿的数目和雷达覆盖的范围半径 d 。接下来 n 行，每行两个整数，表示每个岛屿的坐标位置

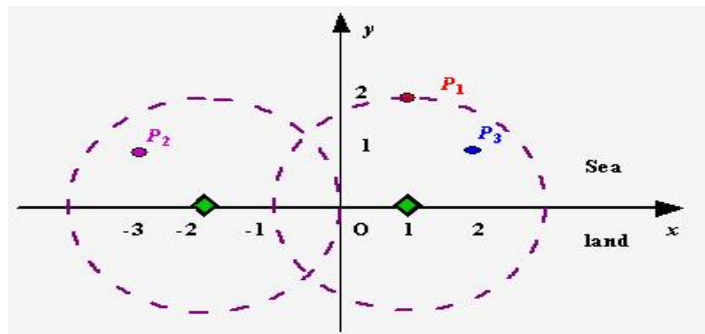
(x,y) 。

➤ 样例输入：

3	2
1	2
-3	1
2	1

➤ 输出：一行一个整数，即能将所有岛屿全部覆盖至少安装的雷达个数，如果无解则输出“-1”。

➤ 样例输出：2



例题：晒衣服（贪心+桶）

- 假设衣服在自然条件下用1的单位时间可以晒干A点湿度。抠门的Smart买了1台烘衣机。使用烘衣机可以让他用1个单位时间使1件衣服除了自然晒干的A点湿度外，还可烘干B点湿度，但在1个单位时间内只能对1件衣服使用。
- N件的衣服因为种种原因而不一样湿，现在告诉你每件衣服的湿度，要你求出弄干所有衣服的最少时间（湿度为0为干）。

例题：晒衣服（贪心+桶）

- 输入：第一行N, A, B; $1 \leq \text{湿度}, A, B \leq 500000$, $1 \leq N \leq 500000$ 。接下来N行，每行一个数，表示衣服的湿度。
- 输出：一行，输出弄干所有衣服的最少时间。
- 输入样例：

5	3	1
5		
7		
2		
10		
2		

输出样例：3

例题：智力大冲浪

- 比赛时间分为 n ($n \leq 5000$) 个时段，给出了很多小游戏，每个小游戏都必须在规定期限 t_i ($1 \leq t_i \leq n$)。如果一个游戏没能在规定期限前完成，则要从奖励费 m 元中扣去一部分钱 w_i ， w_i 为自然数，不同的游戏扣去的钱是不一样的。当然每个游戏本身都很简单，保证每个参赛者都能在一个时间段内完成，而且必须从整时段开始。主持人只是想考考每个参赛者如何安排组织自己做游戏的顺序。作为参赛者，小伟很想赢得冠军，当然更想赢取最多的钱！

例题：智力大冲浪

- 输入：第一行为m，表示一开始奖励给每位参赛者的钱；第二行为n，表示有n个小游戏；第三行有n个数，分别表示游戏1到n的规定完成期限；第四行有n个数，分别表示游戏1到n不能在规定期限前完成的扣款数。

➤ 输入样例： 10000 输出样例： 9960

7

4 2 4 3 1 4 6

40 70 10 50 30 60 20

例题：游戏通关（贪心+并查集）

- 小明需要完成 N 个任务才能将这个游戏的通关。
- 每个任务完成时限 T ，就是这个任务必须在时间 T 之前完成（你可以认为游戏刚开始的时间为1），还有完成这个任务小明可以获得一定的奖励 W 。由于小明娴熟的技术以及任务的简单，他可以在一个单位时间将任务完成。
- 小明想要在老师到来之前将任务全部完成，同时他也想获得最多的奖励。

例题：游戏通关（贪心+并查集）

- 第一行有一个整数N，表示需要完成的任务数目；
 - 接下来N行，每行两个整数T，W（中间用一个空格隔开），分别表示完成这个任务的最后期限和完成这个任务后获得的奖励。
- $N \leq 200000$, $T_i \leq 200000$, $W_i \leq 2000$
- 输出数据有且仅有一行，只包含一个整数S，表示最多获得的奖励。
- | ➤ 输入 | 输出 |
|------|----|
| 5 | 15 |
| 2 3 | |
| 1 2 | |
| 4 5 | |
| 1 3 | |
| 3 4 | |

例题：关押罪犯（贪心+并查集）

- S城现有两座监狱，一共关押着 N 名罪犯，编号分别为 $1 \sim N$ 。现在有 M 对罪犯之间积怨已久，如果客观条件具备则随时可能爆发冲突。我们用“怨气值”（一个正整数值）来表示某两名罪犯之间的仇恨程度，怨气值越大，则这两名罪犯之间的积怨越多。如果两名怨气值为 c 的罪犯被关押在同一监狱，他们俩之间会发生摩擦，并造成影响力为 c 的冲突事件。
- 在详细考察了 N 名罪犯间的矛盾关系后，警察局长觉得压力巨大。他准备将罪犯们在两座监狱内重新分配，以求产生的冲突事件影响力都较小，应如何分配罪犯，才能使冲突事件的影响力最大值最小，这个最小值是多少？

例题：关押罪犯（贪心+并查集）

- 第一行为两个正整数 N, M ，分别表示罪犯的数目以及存在仇恨的罪犯对数。接下来的 M 行每行为三个正整数 $a[i], b[i], c[i]$ ，表示 $a[i]$ 号和 $b[i]$ 号罪犯之间存在仇恨，其怨气值为 $c[i]$ 。
- 数据保证 $1 < a[i] \leq b[i] \leq N, 0 < c[i] \leq 10^9$ ，且每对罪犯组合只出现一次， $N \leq 20000, M \leq 100000$ 。

- 冲突事件的影响力最大的最小值。如果本年内监狱中未发生任何冲突事件，请输出0。

输入	输出
4 6	3512
1 4 2534	
2 3 3512	
1 2 28351	
1 3 6618	
2 4 1805	
3 4 12884	

例题: **Ksum** (贪心+堆)

- Peter喜欢玩数组。NOIP这天，他从Jason手里得到了大小为 n 的一个正整数数组。Peter求出了这个数组的所有子段和，并将这 $n(n+1)/2$ 个数降序排序，他想知道前 k 个数是什么。
- 输入格式
第一行包含两个整数 n 和 k 。
接下来一行包含 n 个正整数，代表数组。
- $a[i] \leq 10^9$, $k \leq n(n+1)/2$, $n \leq 100000$, $k \leq 100000$

例题: **Ksum** (贪心+堆)

➤ 输出格式

输出k个数，代表降序之后的前k个子段和。

➤ 样例输入：

3 4

1 3 4

➤ 样例输出

8 7 4 4

例题：优惠券（贪心+堆）

- 新学期开学了，Smart准备买一些新书，书店有 N 本新书，第 i 本新书价格为 P_i ，使用优惠券购买第 i 本书时价格会降为 C_i 。Smart有 K 张优惠券，每本新书只能使用一次优惠券。Smart想知道花不超过 M 的钱最多可以买多少本新书？
- 第一行三个用空格隔开的整数 N, K 和 M ；
- 接下来 N 行，每行两个整数 P_i 和 C_i 。 $1 \leq N \leq 50000$, $1 \leq K \leq N$, $1 \leq P_i \leq 10^9$, $1 \leq C_i \leq P_i$, $1 \leq M \leq 10^{14}$ 。
- 输出一个整数，表示Smart最多可以购买的新书数量。

例题：优惠券（贪心+堆）

➤ 样例输入

4 2 13

4 2

2 1

6 3

9 4

➤ 样例输出

4

例题：探讨人生（贪心+枚举）

- Smart每次他与好友A探讨人生要花费 a 个小时，并可以得到 x 点人生经验；每次与好友B探讨人生要花费 b 个小时，并得到 y 点人生经验。但是Smart的精力是有限的，他只能抽出 n 个小时来跟他的友人们探讨人生，若这 n 个小时并没有被用完，则Smart会把剩下的时间拿来跟好友C聊天，而这并不能得到人生经验。现在Smart想知道，他最多可以得到多少点人生经验。

例题：探讨人生（贪心+枚举）

➤ 输入共1行，包含5个整数n、x、y、a、b，每两个整数之间用一个空格隔开。

➤ 样例1输入

999999999

999999998 5 999999999 5

➤ 样例2输入

999999999

10000 494999 2 99

➤ 输出共1行，包含1个整数，表示Smart最多能得到的人生经验。

➤ 样例1输出

999999998

➤ 样例2输出

4999999994999

例题：数字游戏（贪心+DP）

- 小W发明了一个游戏，他在黑板上写出一行数字 a_1, a_2, \dots, a_n ($n \leq 200$)，然后给你 m 个回合的机会，每个回合你可以从中选一个数擦除它，接着剩下来的每个数字 a_i 都要递减一个值 b_i 。如此重复 m 个回合，所有你擦除的数字之和就是你得到的分数。
- 编程帮小W算算，对于每个给出的 a_n 和 b_n 序列，可以得到的最大得分是多少？

例题：数字游戏（贪心+DP）

➤ 样例输入

3

3

10 20 30

4 5 6

➤ 样例输出

47

例题：马步距离（贪心+搜索）

- 在国际象棋和中国象棋中，马的移动规则相同，都是走“日”字，我们将这种移动方式称为马步移动。如图所示。
- 从标号为 0 的点出发，可以经过一步马步移动达到标号为 1 的点，经过两步马步移动达到标号为 2 的点。任给平面上的两点 p 和 s ，它们的坐标分别为 (x_p, y_p) 和 (x_s, y_s) ，其中， x_p, y_p, x_s, y_s 均为整数。假设棋盘充分大，并且坐标可以为负数。现在请你求出从点 p 到点 s 至少需要经过多少次马步移动？

4	1	2	1	4
1	2	3	2	1
2	3	0	3	2
1	2	3	2	1
4	1	2	1	4

例题：马步距离（贪心+搜索）

- 输入：只包含4个整数，它们彼此用空格隔开，分别为 x_p , y_p , x_s , y_s 。并且它们的都小于10000000。
- 输出：含一个整数，表示从点p到点s至少需要经过的马步移动次数。

分析

- 由于数据范围很大，所以不能直接BFS，首先可以发现从p到s相当于从(0,0)到 $(|x_p - x_s|, |y_p - y_s|)$ 且当p和s相距很远时马步移动的方式只有从(x,y)到(x+1,y+2)或(x+2,y+1)，所以可以先用这个性质将(x,y)减小到可行的范围内，再进行BFS。

例题：切蛋糕（贪心+搜索）

- Facer今天买了 n 块蛋糕，不料被信息组中的同学发现了，没办法，只好浪费一点来填他们的嘴巴。他答应给每个人留一口，然后量了量每个人口的大小。Facer有把刀，可以切蛋糕，但他不能把两块蛋糕拼起来，但是他又不会给任何人两块蛋糕。现在问你，facer怎样切蛋糕，才能满足最多的人。（facer的刀很强，切的时候不会浪费蛋糕）。

例题：切蛋糕（贪心+搜索）

- 输入格式：第一行n，facer有n个蛋糕。接下来n行，每行表示一个蛋糕的大小。再一行一个数m，为信息组的人数，然后m行，每行一个数，为一个人嘴的大小。 $(1 \leq n \leq 50, 1 \leq m \leq 1024)$
- 输出格式：一行，facer最多可以填多少张嘴巴。
- 输入样例：
4 30 40 50 25 10 15 16 17 18 19 20 21 25 24 30
- 输出样例：
7

分析

- 贪心地想，如果一块蛋糕能满足口较大的人，那么口较小的人一定能满足，并且能满足的人可能会更多，所以先满足口较小的人。
- 二分可以满足的最多的人有mid个（已经从小到大排好序了）
- dfs判断n块蛋糕是否满足前mid人是否可行？

分析

- 若所有的蛋糕都不能满足某一个人的口，那么可以将其删去
- 对于一块蛋糕，将其分出去一部分满足一些人后，倘若剩余部分连口最小的人都无法满足，那么剩余部分一定不能满足任何人，就属于浪费的部分。若蛋糕有用的部分（总数-浪费）不能满足前mid个人，那么当前二分的答案不可行
- 在dfs过程中，可能会遇到这么一种情况，第i个人与第i-1个人的口相同，那么考虑，若不能满足第i-1个人，那么一定不能满足第i个人。