



参数搜索的应用

芜湖市第一中学 汪汀



引言

参数搜索是解决最优解问题的一种常见的方法。其本质就是对问题加入参数，先解决有参数的问题，再不断调整参数，最终求得最优解。下面通过几个例子来说明这一点。



问题一 分石子问题

- ✦ 有 N 个石子，每个石子重量 Q_i ；
- ✦ 按顺序将它们装进 K 个筐中；
- ✦ 求一种方案，使最重的筐尽量轻。



问题一 分石子问题

$N=9, K=3$

9 7 5 6 8 4 3 2 7

16 19 16

最大为 19

9 7 5 6 8 4 3 2 7

21 18 12

最大为 21



问题一 分石子问题——分析

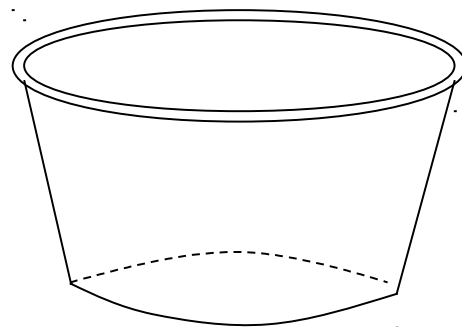
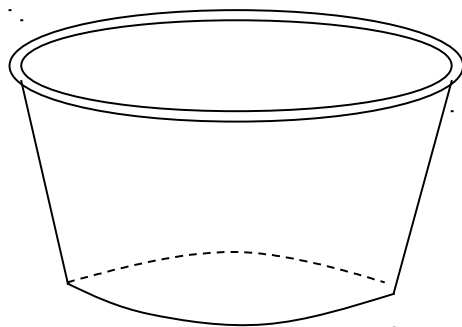
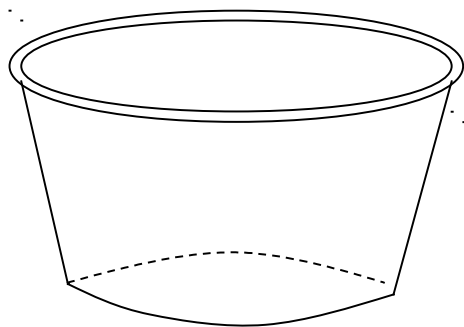
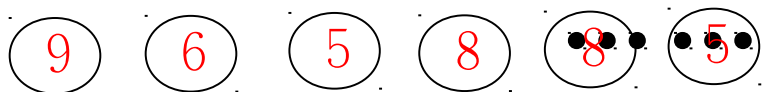
本题可采用动态规划

时间复杂度 $O(N^2)$ 太高了

能否找到实现更简单，更优秀的算法呢？



问题一 分石子问题——分析





问题一 分石子问题——分析

引入参数 P ，求一个判定可行解问题：

判断是否存在最大重量不超过 P 的方案



问题一 石子问题——分析

可以用贪心法解决，具体方法如下：

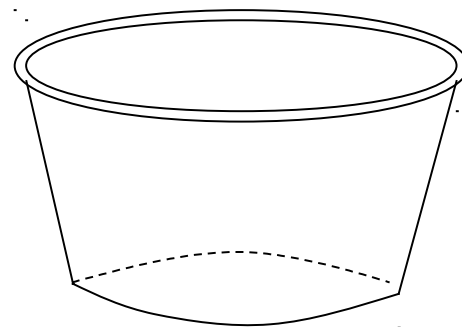
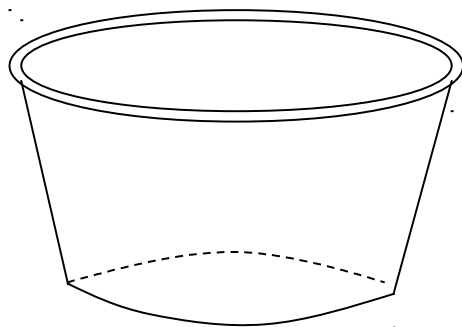
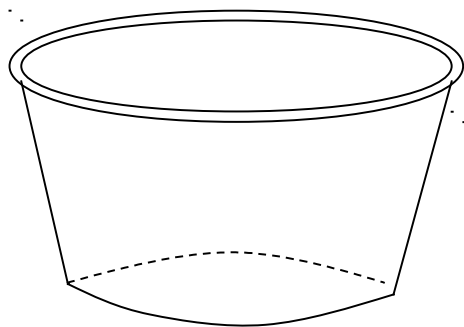
按顺序把石子放入筐，若一个筐中石子总重量不足 **P**，我们就继续对这个筐加入新的石子；如果超过 **P**，则我们
将石子放入新的筐中。



问题一 分石子问题——分析

$N=5, K=3, P=12$

9 6 5 8 3





问题一 分石子问题——分析

回到最初的问题：

从小到大枚举 P ，找到第一个可行解，
该解即为问题所求

令 $T = \sum Q_i$ ，则这个算法的时间复杂度为 $O(TN)$

需要进一步优化！



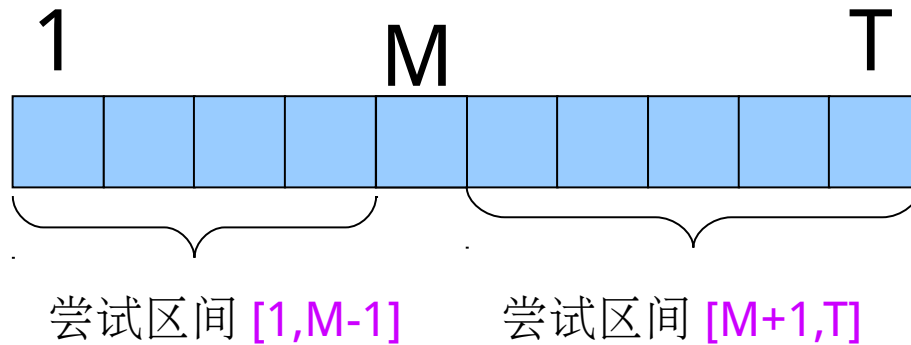
问题一 分石子问题——分析

若我们能找到一个最大重量不超过 P 的方案，
则我们可以找出一个最大重量不超过 $P+1$ 的方案

二分法！



问题一 石子问题——分析



不断重复以上步骤即可找到问题的最优解。

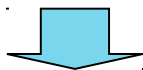
时间复杂度 $O(N \log T)$

特别地，由于答案必定为某一段连续石子的重量和
所以可以得到一个时间复杂度为 $O(N \log^3 N)$ 的算法

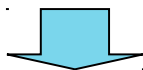


小结

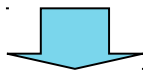
首先引入参数 P ，解决带参数的问题；



用贪心法可以判断是否存在最大重量和不超过 p 的方案；



枚举 P 求出问题的最优解；



二分法降低了算法的时间复杂度，最终解决了问题。



小结

I 首先引入参数 P ，解决带参数的问题；
判断是否存在结果优于 P 的方案

II 调整 P 得到最优解。

通过二分法或迭代法减少调整次数，降低算法时间复杂度。



问题二 最大比率问题

- 有 N 道题目，每道题目有不同的分值和难度，分别为 A_i ， B_i （分值及难度均为正数）；
- 要求从某一题开始，连续选至少 K 道题目，
满足分值和与难度和的比最大。



问题二 最大比率问题

例如

$N=7, K=3$

A 4 3 9 8 9 9 1

B 2 2 1 1 2 1 3

选择第 3 题到第 6 题，比为 $(9+8+9+9)/(1+1+2+1)=7$

这是上例的最优解



问题二 最大比率问题

先来解决一个与之类似的简化版问题:

在一个数列中找连续多个数（至少 K 个），总和最大。

建立一个队列 Q ，开始时队列只有 K 个元素，按顺序往队列中添加新的元素，添加后计算 $Q_1+Q_2..+Q_{L-K}$ 的和，记为 S 。若 $S < 0$ ，则将 $Q_1, Q_2..Q_{L-K}$ 从队列中删除，否则暂时保留这些数。并不断更新最大值。



问题二 最大比率问题——分析

扫描一遍的复杂度是线性的

这就是最优解

和为 -1 和为 2 和为 -1 和为 12



问题二 最大比率问题——分析

现在我们已经能在 $O(N)$ 的时间内解决简化后的问题了。

这个方法能不能应用于原题？

很遗憾，由于原题中每个数据有**两个参数**，故它们对最终结果产生的影响是不确定的，因此对于原题我们不能直接套用这个算法。

现在必须消除这个不确定因素。



问题二 最大比率问题——分析

为此，我们必须引入参数 p ，求一个判定可行解问题：

判断是否存在一个比大于 p 的方案



问题二 最大比率问题——分析


新的问题可以这样描述:

求两个下标 i', j' , 满足

$$j' - i' + 1 \geq k \quad ①$$

$$(A_{i'} + A_{i'+1} \dots + A_{j'}) / (B_{i'} + B_{i'+1} \dots + B_{j'}) > p \quad ②$$

$$② \Rightarrow A_{i'} + A_{i'+1} \dots + A_{j'} > p(B_{i'} + B_{i'+1} \dots + B_{j'})$$


$$(A_{i'} - pB_{i'}) + (A_{i'+1} - pB_{i'+1}) \dots + (A_{j'} - pB_{j'}) > 0$$



问题二 最大比率问题——分析

$$(A_{i'} - pB_{i'}) + (A_{i'+1} - pB_{i'+1}) \dots + (A_{j'} - pB_{j'}) > 0$$

$$\text{令 } C_i = A_i - pB_i, \quad C_{i'} + C_{i'+1} \dots + C_{j'} > 0$$



可以借用刚才的结论

在数列 C 中找一段连续的数（至少 K 个），和为正数。

我们能在 $O(N)$ 的时间内找到中和最大的一段，记为 S ：

- 若 $S > 0$ 则找到了一个可行方案
- 若 $S \leq 0$ ，则问题无解



问题二 最大比率问题——分析

$O(N)$ 的时间判断出是否存在比不小于 P 的方案。

通过二分法，调整参数 P 的大小，不断逼近最优解。

时间复杂度 $O(N \log T)$



小结

上例的难点在于每道题有难度和分值两个数据，使我们无法确定它们对最终结果的影响，因此不能直接套用之前的扫描法。

引入参数后，成功的消除了这个不确定因素，从而轻松的解决了问题。



总结

➤ 回顾两个例子：

➤ 分石子问题：

动态规划	时空编程复杂度都很高
参数搜索	时空复杂度比较优秀

效率比较高

➤ 最大比率问题：

直接扫描	无法解决
参数搜索	问题豁然开朗

降低思维复杂度

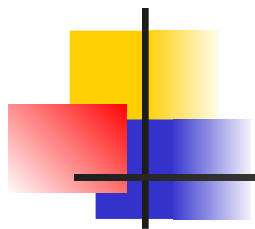


总结

- 参数搜索主要解决最优解问题，它的应用非常广泛，优点也十分突出，使得我们解此类问题又多了一种有力的武器。

- 通常情况下，它都不是独立出现的，需要配合其他算法。例如：引入参数后往往要用贪心，动态规划等算法解决判定可行性问题，而为了减少枚举次数常采用二分等方法。

- 形式较为简单，但实际应用中，我们不能拘泥于形式化的东西，必须根据实际情况，灵活使用，大胆创新，这样才能游刃有余的解决问题。



谢谢