



北京大学
PEKING UNIVERSITY

制作菜品

林泽辉

2020 年 8 月 19 日

1. 题目大意

2. 得分情况

3. 题目解答

3.1 $m = n - 1$

3.2 $m \geq n - 1$

3.3 $m \geq n - 2$

4. 算法实现与得分

5. 拓展与应用

6. 参考文献

题目大意



给定 n 种原材料的质量 $\{d_i\}_{i=1}^n$ ，制作 m 道菜品。所有原材料质量之和 $\sum\{d_i\}_{i=1}^n$ 等于 $m \cdot k$ 。一种原材料可用于多道菜品。上述 n 、 m 、 d_i 、 k 均为正整数。问是否存在满足以下条件的制作方案：

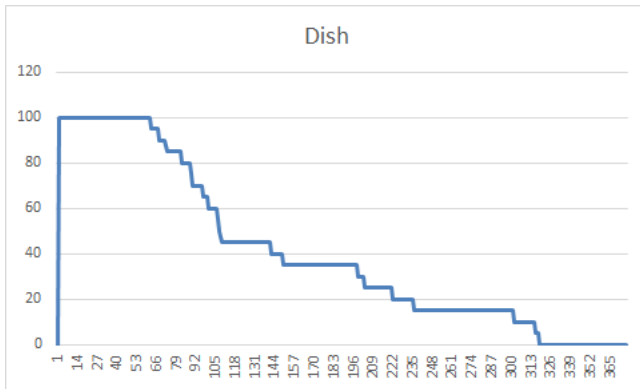
- ▶ 每道菜最多使用 2 种原材料；
- ▶ 每道菜恰好使用 k 克的原材料；
- ▶ 每道菜使用的每种原材料的质量为正整数克；
- ▶ 所有原材料都恰好用完。

若存在满足条件的制作方案，则给出一种具体的制作方案。

得分情况



北京大学
PEKING UNIVERSITY



题目解答



北京大学
PEKING UNIVERSITY

- ▶ 当 $n \leq 4$ 且 $m \leq 4$ 时，使用朴素的搜索算法，每次枚举 1 种或 2 种原材料做一道菜即可。期望得分 15 分。
- ▶ 接下来我们考虑 n 和 m 更大的情况。

$$m = n - 1$$



- ▶ 相信大家都注意到了 $m = n - 1$ 这个特殊条件。
- ▶ 使用 2 种原材料做 1 道菜非常简单，直接做就行了。
- ▶ 使用 3 种原材料做 2 道菜：显然最少的那种原材料不够做 1 道菜；随便再找一种原材料，与这种原材料做成一道菜，问题就变成了 2 种原材料做 1 道菜，或 1 种原材料做 1 道菜，也不困难。
- ▶ 使用 n 种原材料做 $n-1$ 道菜：同样，最少的那种原材料不足以做 1 道菜。可以证明最少的那种原材料与最多的那种原材料结合，一定足以做 1 道菜，并且最多的那种原材料还有剩。这样的话，问题就变成了 $n-1$ 种原材料做 $n-2$ 道菜。
- ▶ 时间复杂度 $\Theta(n \log n)$ ，期望得分 10 分，与朴素的搜索算法结合可得 25 分。

$$m \geq n - 1$$



- ▶ 显然 $m \geq n$ 时，最多的原材料的质量一定大于等于 k 。我们直接使用最多的那种原材料做成一道菜。原材料数量不变或减 1，需要做的菜减 1。要么最终剩下 1 种原材料，要么变为 $m = n - 1$ 的情况。
- ▶ 与前面的 $m = n - 1$ 的方案结合，可以解决 $m \geq n - 1$ 的情况。时间复杂度 $\Theta(m \log n)$ ，期望得分 20 分，与朴素的搜索算法结合可得 35 分。

$$m \geq n - 2$$



- ▶ 在前面算法的基础上写搜索：每次枚举与最少的原材料组合的另一种原材料。时间复杂度 $O(n!)$ ，期望得分 45 分。
- ▶ 前面已经解决了 $m \geq n - 1$ 的情况，我们自然希望 $m = n - 2$ 也能够利用前面的方案解决。
- ▶ 一个很直接的想法是，能否找到一对数加起来恰好为 k ？如果能找到，那么这一对数做一道菜，剩下的就是 $m = n - 1$ 的情况了，可以直接套用前面的方案解决。
- ▶ 更进一步地，能否找到 x 个数，它们的和恰好为 $(x - 1) \cdot k$ ？这样就可以使用这 x 种原材料制作 $x - 1$ 道菜，再使用剩下的 $n - x$ 个数制作剩下的 $n - x - 1$ 道菜了。实际上，这是 $m = n - 2$ 时有解的充分必要条件。

$$m \geq n - 2$$



定理 1

$m = n - 2$ 的问题有解的充分必要条件为它可被划分为两个 $m = n - 1$ 的问题。

- ▶ 举个例子：7 种原材料做成 5 道菜，每道菜使用 $k = 6$ 个单位的原材料。可以使用 1 2 3 做第 1 和第 2 道菜，使用 4 5 6 7 做剩下的菜。

菜 1	菜 2	菜 3	菜 4	菜 5
1	2	4	5	6
2	2	4	5	7
2	2	4	5	7
2	3	4	6	7
2	3	5	6	7
2	3	5	6	7

$$m \geq n - 2$$



定理 1

$m = n - 2$ 的问题有解的充分必要条件为它可被划分为两个 $m = n - 1$ 的问题。

证明.

- ▶ 由前面的方案知，若能划分则必然有解。充分性得证。
- ▶ 必要性可由归纳法证明。
- ▶ 这里介绍一个巧妙的证明方法：把原材料看成点，那么做一道菜就是在它所用的原材料之间连边。 n 个点 $n - 2$ 条边，连成图后，图一定至少能分为 2 个部分。因此原 $m = n - 2$ 的问题若有解，一定能划分成两个 $m = n - 1$ 的子问题。必要性得证。



$$m \geq n - 2$$



于是我们考虑如何寻找 $m = n - 2$ 的划分，这是一个 NPC 问题。

- ▶ 搜索算法：在 $\{d_i\}_{i=1}^n$ 中搜索一个大小为 x 、和为 $(x-1) \cdot k$ 的子集。时间复杂度 $\Theta(2^n)$ 。
- ▶ 动态规划算法：用 $f[i][j][s]$ 表示前 i 种原材料，是否存在大小为 j ，和为 s 的子集。

初始化： $f[0][0][0]$ 为 true，除此之外 $f[0][*][*]$ 为 false。

状态转移： $f[i][j][s] = f[i-1][j][s] \mid f[i-1][j-1][s-d_i]$ 。

若存在一个 x 使得 $f[n][x][(x-1) \cdot k]$ 为 true，则问题有解，再利用动态规划算法过程中记录的状态还原出划分的方案。

时间复杂度 $\Theta(n^3 k)$ 。

$$m \geq n - 2$$



于是我们考虑如何寻找 $m = n - 2$ 的划分，这是一个 NPC 问题。

- ▶ 动态规划算法改进：注意到目标是大小为 x 、和为 $(x - 1) \cdot k$ 的子集，可以把每个数都减去 k ，目标变成了找一个和为 $-k$ 子集。

即：令 $d'_i = d_i - k$ 。用 $f[i][s]$ 表示 $(d'_1, \dots, d'_i)$ 是否存在和为 s 的子集。

初始化： $f[0][0]$ 为 true，除此之外 $f[0][*]$ 为 false。

状态转移： $f[i][s] = f[i-1][s] \mid f[i-1][s-d'_i]$ 。

若 $f[n][-k]$ 为 true，则问题有解，再利用动态规划算法过程中记录的状态还原出划分的方案。

时间复杂度 $\Theta(n^2 k)$ 。

算法实现与得分



北京大学
PEKING UNIVERSITY

- ▶ 15 分: $n \leq 4$ 且 $m \leq 4$, 朴素的搜索即可。
- ▶ 20 分: $m \geq n - 1$, 使用题解中的方案求解, 时间复杂度 $\Theta(n \log n)$ 。
- ▶ 35 分: 将 15 分与 10 分算法结合。
- ▶ 45 分: $n \leq 10$ 且 $m \leq 10$, 搜索时, 固定使用质量最少的原材料, 枚举与其组合的另一种原材料。时间复杂度 $O(n!)$ 。
- ▶ 60 分: 使用 $\Theta(2^n)$ 的搜索算法。
- ▶ 70 分: 直接使用 $\Theta(n^3 k)$ 的动态规划算法。
- ▶ 85 分算法 1: 使用 bitset 优化 $\Theta(n^3 k)$ 的动态规划算法。
- ▶ 85 分算法 2: 直接使用 $\Theta(n^2 k)$ 的动态规划算法。
- ▶ 100 分: 使用 bitset 优化 $\Theta(n^2 k)$ 的动态规划算法。



在定理1中我们证明了 $m = n - 2$ 的问题有解当且仅当它能够被划分为 2 个 $m = n - 1$ 的问题。

可以将结论推广为: $m = n - p$ 的问题有解当且仅当它能够被划分为 p 个 $m = n - 1$ 的问题 ($1 \leq p \leq n/2$)。

此题的 $m = n - 1$ 的方案可用于解决一个经典问题：如何快速对离散概率分布进行采样。即给定 n 个数和它们出现的概率 $\{p_i\}_{i=1}^n$ ，每次随机生成一个数，它是给定的数中的第 i 个的概率为 p_i ，要求快速随机生成若干个数。

使用题解中给出的 $m = n - 1$ 的解法，可以在 $\Theta(n \log n)$ 的时间内求出 n 个数放入 $n - 1$ 个大小一致的盒子、每个盒子最多有两个数的方案。生成随机数时，先均匀随机一个盒子，再均匀随机盒子内的一个位置，取盒子中对应位置的数即可。此算法名为 alias method。A. J. Walker 于 1974 年提出了这个巧妙的算法 [1][2]。Vose 于 1991 年将它改进为线性的做法 [3]。线性的做法不是非常复杂，有兴趣的同学可以自行思考。

- [1] Alastair J Walker. New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters*, 10(8):127–128, 1974.
- [2] Alastair J Walker. An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):253–256, 1977.
- [3] Michael D Vose. A linear algorithm for generating random numbers with a given distribution. *IEEE Transactions on software engineering*, 17(9):972–975, 1991.



北京大学

PEKING UNIVERSITY

谢谢!