背包

首先，所使用的背包容量肯定越大越好，所以先对背包容量排个序。

然后考虑一个很简单的 dp，设 $f_{i,j}$ 表示用前 i 个背包，是否可以装好状态为 j 的物品。枚举子集暴力转移，复杂度 $O(m3^n)$，可以拿到 50 分。

实际上枚举子集这一转移方法很费时，考虑枚举单个物品转移。设 $f_{i,j}$ 表示用前 i 个背包，装状态为 j 的物品后，第 i 个背包最多还剩多少容量，然后枚举单个物品即可。复杂度 $O(n^2 2^n)$，实际常数会更小。

移动细胞

考虑设计状态 $f_{i,j}$ 表示前 i 列的黑色细胞已联通，且第 i 列黑色细胞的开头在第 j 行的答案。每次转移由 $f_{i-1,k}$ 转移过来，其中 k 在一段区间内。直接转移是 $O(n^2m)$，用单调队列维护即可 $O(nm)$。

异或和

显然我们可以理解为两个数 $C = A\ xor\ B$ 是定值，然后可以调整某一些位置，使得 $X(A) + X(B)$ 尽可能大。

那么考虑一下我们可以怎么调整。

如果某一位 C 是 1，那么调整没啥用，而如果 C 某一位是 0，则我们希望可以通过调整使得 A, B 对应位置都为 1。由于 A, B 的异或值是定值，所以只需要调整 A。把 A 序列全部异或起来，能调整的值就是 $A_i\ xor\ B_i$ 所构成的线性基。

从高到低贪心即可。

序列

不妨从后往前选数，这样 a_i 变成了单调不降。

设 $f_{i,j}$ 表示目前有 j 个值为 i 的数的答案。那么加入 a_i，就有 $f_{a_i,j} - b_i + c_{a_i} \to f_{a_i,j+1}$。

再考虑合并，从 i 合并到 $i+1$ 会有 $f_{i,j} + \lfloor \frac{j}{2} \rfloor \times c_{i+1} \to f_{i+1,\lfloor \frac{j}{2} \rfloor}$。

由于 a_i 单调不降，所以以后不会再出现小于 a_i 的数。因为每次向上合并时个数都会除以二，所以合并的复杂度是 $O(n)$ 的，总复杂度 $O(n^2)$。