多重背包问题
二进制拆分优化

主讲人：邓哲也

多重背包问题

有 N 件物品和一个容量为 V 的背包。

第 i 件物品的费用是 c[i]，价值是 w[i]。

第 i 件物品最多有 a[i] 个可用。

求将哪些物品装入背包可以使得价值总和最大。

多重背包问题

定义状态 f[i][j] 表示从前 i 件物品中选出容量为 j 的背包能获得的最大价值。

思考转移方程。

多重背包问题

$f[i][j] = \max(f[i-1][j-k \cdot c[i]] + k \cdot w[i] \mid 0 \leqslant k \leqslant a[i])$

这样的时间复杂度已经到了 $O(NV \cdot \sum a[i])$

物品拆分

比如一个物品有最多有15个可用，15的二进制是1111

那可以把这个物品拆成4份，分别代表8个物品、4个物品、2个物品、1个物品。

这样不管选出0~15个中的任何数量个，都可以用这4份组合出来。

物品拆分

再比如一个物品有最多有12个可用，12的二进制是1100

拆成两份，分别代表8个和4个并不能满足要求。

因此对于任何a[i]，我们都先拆成1个、2个、4个、…、2^k个，k 是满足 $2^k-1<n$ 的最大值。

此时可以满足可以拼出0~2^k-1中的任何数量。

只要再加一份表示 $n-2^k+1$个物品即可。

比如 12 就可以拆分成 1个、2个、4个、5个。

多重背包问题

这样对于一个最大可用数为 a[i] 的物品来说，我们就把它拆分成了 O(log a[i]) 个只能用一次的物品，体积为 kc[i]，价值为 kw[i]。（k 表示倍数）

这样问题就变成了 01 背包问题，只是物品数量变成了 O(Σ(log a[i])) 个。

时间复杂度是 O(VΣ(log a[i]))

下节课再见