# 多重背包问题
# 单调队列优化

主讲人：邓哲也

# 单调队列优化

对于 f[i][j]，令 b[i] = min(a[i], j / c[i])

f[i][j] = max(f[i-1][j-k•c[i]]+k•w[i] | 0≤k≤b[i])

观察 f[i][j] 会从哪些状态转移过来。

对于 {j-k•c[i] | 0≤k≤b[i]} 这些数模 c[i] 都是同余的。

我们令 mod = j % c[i], div = j / c[i], 那么 j = div * c[i] + mod

f[i][j] = max(f[i-1][mod + (div - k)c[i]]+k•w[i] | 0≤k≤b[i])

# 单调队列优化

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

用 div-k 替换掉 k

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

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

# 单调队列优化

f[i][j] = max(f[i-1][mod + k•c[i]]-k•w[i] | div-b[i]≤k≤div) + div•w[i]

考虑 {mod, mod+c[i], mod+2c[i], mod+3c[i], …, j}

f[i][j] 就是求 j 前面的 (b[i]+1) 个数对应的 f[i-1][mod+kc[i]]-kw[i] 的最大值。

对于最外层 i 的枚举，b[i]+1 是固定的。

问题转化为求一个固定长度的滑窗内的最大值。

# 单调队列优化

维护一个单调下降的队列。

每次加入的时候加入队尾，保证队头到队尾单调递减。

弹出队头，直到满足队头在滑窗内。

队头的值就是这个滑窗内的最大值。

这样这一步就是线性的。

我们枚举 mod，再枚举 div，求 f[i][j] 是用单调队列，总共是 O(V) 的。

总的时间复杂度仍然保持在了 O(NV)。

下节课再见