G/R题题解

**题意：**

裸多重背包外加一个体积可变，且体积和收益呈二次关系的物品，问背包能获得最大价值是多少。

**题解：**

1.多重背包

首先关注多重背包，多重背包的二进制优化在背包九讲中有详细的描述，这里不再赘述。

这里主要关注多重背包的单调队列优化，为什么会想到用单调队列优化，我们先来看如下的例子，假设某个物品体积为3，数量为3，背包容量为20。

更新顺序如下：

0，1，2，无法更新

3，由0更新

4，由1更新

5，由2更新

6，由0，3更新

7，由1，4更新

8，由2，5更新

9，由0，3，6更新

发现1只会用来更新4，7，10；2只会用来更新5，8，11，等等

例如10，我们只需要在1，4，7中取得转移，是不是和滑动窗口很像？但是还有一点问题，4用1更新时是用1+val，而7用1更新时是1+2val，10是1+3val，针对这个问题，我们可以假设当前更新的背包重量为W=k\*w[i]+b，b是W对w[i]取模的剩余系中的一个元素，然后把W对应的最大价值dp[W]减去k倍的当前物品价值放入优先队列。出队的时候将出队元素加上当前k倍的价值

为什么可以这么做？看以下式子

5由2+val[i]更新而来

7由2+2\*val[i]，5+val[i]更新而来，假设我们要在出队时加上2\*val[i]，5就应该在入队时减去(k=1)\*val[i]，而2应当减去(k=0)\*val[i]，7入队时应减去(k=2)\*val[i]，以方便10在计算时加上3\*val[i]。

说了这么多，其实转移方程一句话就能概括

单调队列优化的多重背包时间复杂度

2.神奇物品

由于m较小，枚举每个神奇物品的体积并对应更新即可，

神奇物品时间复杂度为

总复杂度为