

# O-宝可梦

## 混合背包问题

单调队列有点难码，这里就使用二进制拆分优化来完成多重背包问题

众所周知，从  $2^0, 2^1, 2^2, \dots, 2^{k-1}$  这  $k$  个 2 的整数次幂中选取若干个数相加，可以表示出  $0 \sim 2^k - 1$  之间的任何整数。进一步的，我们求出满足  $2^0 + 2^1 + \dots + 2^p \leq C_i$  的最大的整数  $p$ ，设  $R_i = C_i - 2^0 - 2^1 - \dots - 2^p$ ，那么通过这  $p + 2$  个数可以表示出  $0 \sim C_i$  间的任意整数，因此我们把数量为  $C_i$  的物品拆分成了  $p + 2$  个物品，再做朴素的多重背包即可。

拆分实现的代码如下：

```
rep(i, 1, n){
    rd(x), rd(y), rd(z); int s = 1;
    for(int k = 0; s <= z; s += 1 << k){
        int cnt = 1 << k;
        val.push_back(1LL * cnt * x);
        vol.push_back(1LL * cnt * y);
        ++k;
        if(s + (1 << k) > z) break;
    }
    if(s < z){
        val.push_back(1LL * (z - s) * x);
        vol.push_back(1LL * (z - s) * y);
    }
}
```

