多重背包问题

- P139 多重背包问题1
- P140 多重背包问题2
- P141 多重背包问题3

朴素法

时间复杂度：`O(nms)`

```
while(n--){
    int v, w, s;
    cin >> v >> w >> s;
    for(int j = m; j >= v; j--)
        for(int k = 1; k <= s && k * v <= j; k++)
            f[j] = max(f[j], f[j - k * v] + k * w);
}
```

二进制分解思想

时间复杂度：`O(nmlogs)`

设第i种物品总数s[i]，将第i种物品分成若干件物品，其中每一件物品有一个系数，这件物品的费用和价值乘以这个系数，系数分别为1，2，4，.....，2^{k-1}，s[i]- 2^k+1，k是满足s[i]- 2^k+1>0的最大整数。

例如：第i种物品有10个，分解为1，2，4，3，k=3。

```
while(n--){
    int v, w, s;
    cin >> v >> w >> s;
    for(int k = 1; k <= s; k *= 2){
        s -= k;
        nodes.push_back({v * k, w * k});
    }
    if(s > 0) nodes.push_back({v * s, w  * s});
 }

for(auto node : nodes){
    for(int j = m; j >= node.v; j--)
        f[j] = max(f[j], f[j - node.v] + node.w);
}
```

单调队列优化

时间复杂度：`O(nm)`

`f[i][j] = max(f[i-1][j], f[i-1][j-v]+w, f[i-1][j-2v]+2w, f[i-1][j-3v]+3w, ..., f[i-1][j-sv]+sw)`


```
while(n--){
        int v, w, s;
        cin >> v >> w >> s;
        memcpy(g, f, sizeof f);

        for(int r = 0; r < v; r++){
            int hh = 0, tt = -1;
            for(int j = r; j <= m; j += v){
                while(hh <= tt && j - s * v > q[hh]) hh++;
                while(hh <= tt && g[q[tt]] + (j - q[tt]) / v * w <= g[j]) tt--;
                q[++tt] = j;
                if(hh <= tt) f[j] = g[q[hh]] + (j - q[hh]) / v * w;
            }
        }
    }
```

状态压缩

- P292 小国王
- P297 玉米田
- P298 [NOI 2001] 炮兵阵地