

# 新年的魔塔 解题报告

杭州学军中学 王逸松

## 1 试题来源

UOJ Goodbye Jiawu E. 新年的魔塔

## 2 试题大意

魔塔是一种固定数值RPG游戏。

这个游戏在一个 $n \times m$ 的棋盘上进行。行编号为 $1, \dots, n$ ，列编号为 $1, \dots, m$ 。

为了简单起见，我们假设这个魔塔只有一层。

我们用 $(x, y)$ 来表示第 $x$ 行第 $y$ 列的格子，

这个格子可能是空地或不可通行的墙壁，也可能有怪物，或者有宝物，或者是门。

玩家有以下几个属性：

- 生命值HP，简称“生命”。
- 攻击力ATK，简称“攻击”。
- 防御力DEF，简称“防御”。
- 魔法防御力INT，简称“魔防”。

在任意时刻，玩家的属性都满足 $HP > 0$ ,  $ATK \geq 0$ ,  $DEF \geq 0$ ,  $INT \geq 0$ ，且这些属性都是整数。

每一种怪物也有以下属性：

- 生命值HP
- 攻击力ATK
- 防御力DEF

这些属性也都是整数且 $HP > 0$ ,  $ATK \geq 0$ ,  $DEF \geq 0$ 。

玩家与一只怪物战斗时，首先玩家的HP 会加上玩家的魔防的值，然后玩家作为攻击方，怪物作为防御方。

攻击方的生命值记为 $HP_0$ ，攻击力记为 $ATK_0$ ，防御力记为 $DEF_0$ ，

防御方的生命值记为 $HP_1$ ，攻击力记为 $ATK_1$ ，防御力记为 $DEF_1$ 。

```
while (true)
{
    damage = max(0, ATK_0 - DEF_1)
    HP_1 = HP_1 - damage
    if (HP_1 <= 0)
        break
    else
        交换攻击方和防御方
}
```

这段伪代码的意思是，每个回合防御方受到“攻击方攻击力”减“防御方防御力”的伤害（小于等于0 则不会受到伤害），然后如果双方HP 都大于0 则交换攻击方和防御方，否则战斗结束。

这时如果玩家的 $HP \leq 0$ ，则游戏结束。否则玩家的 $HP = \min(HP, \text{战斗前玩家的HP})$ 。

有些怪物还有一个或两个特殊属性值 $p, q$ :

$p$ 或 $q$ 的值	属性名称	属性效果
0	普通	无
8	坚固	玩家每回合对怪物造成最多1HP 的伤害
16	先攻	第一个回合前，怪物先对玩家攻击一次
64	魔攻	怪物的攻击无视玩家的防御（即把玩家的DEF 视为0）
$-x$	连击	怪物每回合攻击 $x$ 次（ $x > 1$ ）

保证 $p$  和 $q$  都不等于0 时， $p, q$  表示不同的属性，且不会有一种怪物同时有连击和先攻的属性。

宝物分为以下几种：

- 血瓶：增加一定生命值
- 红宝石：增加一定攻击力

- 蓝宝石：增加一定防御力
- 绿宝石：增加一定魔防
- 钥匙：可以用来开门，开一扇门消耗一把钥匙，其中黄钥匙能开黄门，蓝钥匙能开蓝门，红钥匙能开红门。

玩家从某个位置开始游戏，每次移动只能走到相邻（有公共边）且不是墙壁的格子上。

- 如果这个格子上有怪物，则玩家与这只怪物战斗。
- 如果这个格子上有宝物，则玩家获得这个宝物。
- 如果这个格子是门，则玩家使用相应的钥匙开门。

以上事件结束后，该格子变为空地。

给定玩家的初始状态（位置，HP，ATK，DEF，INT，各种钥匙的数量）和目标状态，要求给出一种玩法，使玩家能达到目标状态的位置且各项属性和钥匙的数量都不小于目标状态的属性和钥匙的数量。在这个条件下，玩家的HP 越大越好。

此题为提交答案型试题。

输入数据见tower.zip。

### 3 算法介绍

#### 3.1 第一个测试点

这组数据规模很小，写一个模拟器手玩就可以了。

#### 3.2 第二个测试点

假设这组数据有 $2n$ 行，一开始的HP为 $m + 1$ 。

容易发现，这组数据由 $n + 1$ 个房间组成，其中目标状态的位置所在的房间里有一个能加 $m - 1$ 的HP值的血瓶和一只ATK为 $m$ ，属性为2连击的怪物（我们称它为Boss）。

对于其他房间，都有若干怪物和若干蓝宝石，打死这些怪物所花费的血量跟该房间里蓝宝石加的防御值相等。

除了Boss房间以外，清空每个房间后都会得到一把黄钥匙。

假设我们进入并清空了若干房间，损失了 $k$  ( $1 \leq k \leq m$ )的HP值，于是打Boss前的HP为 $2m - k$ ，打Boss损血 $2(m - k)$ ，打完Boss后HP为 $k$ 。

这就说明了，这组数据是个 $n$ 个物品，容量为 $m$ 的01背包问题。

第二个测试点中 $n = 100$ ,  $m = 10^7$ ，直接dp就可以过。

#### 3.3 第三个测试点

同第二个测试点，但是 $n = 55$ ,  $m = 8 \times 10^{15}$ ，使用meet in the middle优化搜索即可。

#### 3.4 第四个测试点

有 $n$  ( $n = 24$ )只怪，每只怪后面有若干个宝石和一把黄钥匙。

目标状态要求至少有 $n$ 把黄钥匙，说明这 $n$ 只怪都要打，而且要最小化损血。

考虑状压DP， $dp[S]$ 表示只打死 $S$ 这个集合中的怪，最少的损血量，然后复杂度就是 $O(n2^n)$ 了。

### 3.5 第五个测试点

$n \times n$ 的地图，只有 $k$  ( $k = 10$ )个点上面没有怪，且每只怪的损血是固定的，要求收集所有蓝宝石。

直接用斯坦纳树做就行了，复杂度 $O(n^2 3^k)$ 。

### 3.6 第六个测试点

同上，但是蓝宝石被换成了血瓶。

注意如果吃光所有血瓶，答案不一定是最优的。

### 3.7 第七个测试点

$100 \times 100$ 的地图，到处都是怪，一副不能玩的样子。

仔细观察后发现，一开始身边的怪是无伤的，先打死他们，可以得到一个蓝宝石，接下来又有一些怪无伤，打完后又可以得到一个蓝宝石。

.....

这样就可以清光所有怪了。

### 3.8 最后三个测试点

这是经典的魔塔模型。

考虑一个通用的算法：

记 $f[S]$ 为状态为 $S$ 时的 $max\_hp$ ，于是可以状压DP。

其中 $S$ 为一个bitset，保存每个格子是否已经到达过，转移时只要枚举下一个可以达到的格子即可。

所有状态和转移构成一张拓扑图，只要每次取出bitset中“1”的个数最少的状态，就能只用一个bfs来实现这个DP了。

但是这样做根本没法跑出第八个测试点，需要优化。

#### 优化一

很多格子是空地，很多格子是墙壁，我们只要记录有事件（怪物，宝物，门）的格子的状态即可。

#### 优化二

怪物的属性满足一个性质：对任意怪物，不管玩家处于什么状态，玩家的ATK, DEF, INT中任意一项的增加都不会使打这种怪的损血变多。

于是可以在每次转移后，强制将所有能吃的宝物都吃掉（0损血的怪也可以直接杀掉），这样就能跑出第八个测试点了。

### 优化三

考虑一只怪，如果它边上的所有格子都已经能到达，就说明这只怪打了也是白打，可以直接在bitset中将它标记为已经访问过。

对于门也可以这样做，但是要把钥匙数量也记录在状态里，否则会因算错钥匙的数量而WA掉。

### 优化四

杀怪（或开门）有目的性。

考虑这样的情况，杀了一只怪（或开了一扇门）后，如果下一步能访问到的事件只多了一个，那么从当前状态开始，一定存在一种最优的玩法，下一步访问的是那个事件。

在加了以上所有优化后，直接跑第十个测试数据，不一定能跑出来。

因为那组数据里有很多红门，导致优化三和优化四几乎没有效果。

但是数据里只有两把红钥匙，我们只要枚举打开的红门是哪两个，然后将其他红门看成墙壁，跑之前的算法，就行了。

第8, 9, 10 号测试点的模拟程序 密码：2e2n

（请无视玩家死亡时模拟程序给出的分数）

## 4 参考代码和数据

最后三个测试点的程序：tower.cpp（这个程序多输出了一些调试信息）

输入数据和checker：tower.zip