TSOI2 解题报告

锻炼计划(exercise)

由于时间只有 1440 分钟，所以可以直接模拟
使用一个 1..1440 的线性数组 delta，delta[i]表示第 i 分钟体力的改变量
delta[i]的初始值为 1，表示每分钟体力恢复 1
对于每一个计划的起止时间(t1, t2)，把 delta[t1..t2]的值减去此计划每分钟耗费的体力值
初始体力为 s，依次处理 1~1440 分钟：

- s <- s+delta[i]
- 判断是否死亡

期望得分 100

魔兽争霸(war)

本题考查二叉树的应用

修改 HP 时把原 HP 值在树中删除,再将新的 HP 值插入,复杂度 O(logn)。同时在树的每个节点上维护以该点为根的子树上共有多少个节点，则可以在 O(logn)时间内回答询问。

总复杂度为 O(m*logn)

具体实现请参考标准程序

其他做法：

- 每次询问时调用快速排序，复杂度 O(m*n*logn)。期望得分 30 至 40
- 把快速排序优化，递归时只向包含第 k 大数的方向递归，复杂度O(m*n)。期望得分 60 至 70
- 随时使所有的 HP 值成为一个有序的序列，修改时在序列中找到要修改的 HP，修改之后再移动到合适的位置，使序列保持有序，复杂度也为 O(m*n)

盟军敢死队(cmdo)

解法一：
对每一个敌人赋予一个编号 1~n(n 表示敌人数)
对于表示一个消灭所敌人的顺序的序列 Q，用 p[x]表示敌人 x 在序列中的位置

如果敌人 b 在敌人 a 的视线范围内，则必有 p[a]<p[b]
显然，如果对于所有有看守关系的敌人对(a,b)都满足 p[a]<p[b]，则序列 Q 是一个合法的消灭敌人的顺序
由此可得算法：

- 生成 1~n 这 n 个数的全排列
- 对于每一个排列检查是否对于所有的(a,b)都满足 p[a]<p[b]，如果是则 ans <- ans+1
- 如果 ans=0，输出"Impossible"，否则输出 ans

复杂度 $O(n!*n^2)$
期望得分 90

解法二：
基于状态压缩的动态规划
推荐使用记忆化搜索的方式实现：

- 直观、易读
- 避免计算不必要的状态

使用一个 n 位(bit)的二进制数表示当前已经消灭敌人

- 如共有 5 个敌人，第4,5 个敌人已经消灭，0/1 分别表示未消灭和已消灭，则状态为 00011，$(00011)_{2=}(3)_{10}$，则第 4,5 个敌人已经被消灭的状态表示为 3

定义 f(x)表示从状态 x 开始，消灭剩余所有敌人的方法数
显然 f(0)即为所求答案：消灭所有敌人的方法数
可以递归地计算函数 f(x)：

- total <- 0
- 找出在状态 x 下，剩余的敌人中可以被直接消灭(e.g.不在任何其他敌人视线范围内)的敌人集合 S
- 对于每一个敌人 E∈S，将 x 中对应敌人 E 的二进制位置为 1，得到一个新状态 y(e.g.消灭这个敌人之后的状态)，total <- total + f(y)
- 函数返回 total
- 特别的，如果 $x=2^n-1$，则函数返回 1（边界条件，表示已经消灭了所有敌人）

另外：

- 在递归计算的过程中，计算 f(x)之后，使用一个数组g 将 f(x)的值保存到 g[x]中，当再次调用 f(x)时，则可以直接返回 g[x]的值，以减少计算量。
- 这样的方式称为**记忆化**

复杂度：

- 每个敌人要么已经消灭，要么没有消灭，所以状态的数量为 2^n
- 状态转移复杂度 n^2
- 总复杂度 $O(2^n*n^2)$

期望得分 100

暗黑破坏神(diablo)

动态规划

用 $f[i][j]$表示前 i 项魔法花费 j 个金币能达到的最大效果值

则 $f[i][j]=\max(f[i-1][j-c[i]*k]+w[i][k])$

$\max(f[n][j])$ $0<=j<=m$ 即为问题的解

用 $g[i][j]$记录达到状态 $f[i][j]$需要使第 i 项魔法学到 $g[i][j]$级，则最后可以推出每项魔法学了多少级

复杂度 $O(n*m*p)$