

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

动态规划

mhy12345

July 17, 2016

汉诺塔

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

汉诺塔

三根柱子，移圆盘，保证小的在上，大的在下。

a. 请求出最优移动次数。

b. 请求出最优移动方案。

汉诺塔

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

汉诺塔

三根柱子，移圆盘，保证小的在上，大的在下。

a. 请求出最优移动次数。

b. 请求出最优移动方案。

$$H(1) = 1;$$

$$H(n) = H(n-1) * 2 + 1;$$

$$H(n) = 2^n - 1;$$

越狱

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

越狱

监狱有连续编号为 $1 \dots N$ 的 N 个房间，每个房间关押一个犯人，有 M 种宗教，每个犯人可能信仰其中一种。

如果相邻房间的犯人的宗教相同，就可能发生越狱，求有多少种状态可能发生越狱

越狱

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

越狱

监狱有连续编号为 $1 \dots N$ 的 N 个房间，每个房间关押一个犯人，有 M 种宗教，每个犯人可能信仰其中一种。

如果相邻房间的犯人的宗教相同，就可能发生越狱，求有多少种状态可能发生越狱

$$dp_1 = M$$

$$dp_i = dp_{i-1} * (M - 1)$$

路径问题

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

路径问题

一个 $N * M$ 的方格图，每条边是一个路径，求左上角走到右下角最短距离的路径条数。

一些点有障碍不能经过。

路径问题

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

路径问题

一个 $N * M$ 的方格图，每条边是一个路径，求左上角走到右下角最短距离的路径条数。

一些点有障碍不能经过。

$f[i][j]$ 表示从位置 $(0,0)$ 走到 (i,j) 方案总数

$$f_{i,j} = f_{i-1,j} + f_{i,j-1} \quad \{(i,j) \text{ is accessible}\}$$

$$f_{i,j} = 0 \quad \{(i,j) \text{ is blocked}\}$$

奶牛沙盘队

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

奶牛沙盘队

农夫顿因开始玩飞盘之后，约翰也打算让奶牛们享受飞盘的乐趣。他要组建一只奶牛飞盘队。

他的 $N(1 \leq N \leq 2000)$ 只奶牛，每只都有一个飞盘水准指数

$R_i(1 \leq R_i \leq 100000)$ 。约翰要选出 1 只或多于 1 只奶牛来参加他的飞盘队。由于约翰的幸运数字是 $F(1 \leq F \leq 1000)$ ，他希望所有奶牛的飞盘水准指数之和是幸运数字的倍数。

帮约翰算算一共有多少种组队方式

奶牛沙盘队

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

奶牛沙盘队

农夫顿因开始玩飞盘之后，约翰也打算让奶牛们享受飞盘的乐趣。他要组建一只奶牛飞盘队。

他的 $N(1 \leq N \leq 2000)$ 只奶牛，每只都有一个飞盘水准指数

$R_i(1 \leq R_i \leq 100000)$ 。约翰要选出 1 只或多于 1 只奶牛来参加他的飞盘队。由于约翰的幸运数字是 $F(1 \leq F \leq 1000)$ ，他希望所有奶牛的飞盘水准指数之和是幸运数字的倍数。

帮约翰算算一共有多少种组队方式

$f[i][j]$ 表示前 i 只奶牛指数之和在模 F 意义下余 j 的方案总数

数字三角形

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

数字三角形

给你一个数字三角形，求从顶端走到底端最大收益

数字三角形

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

数字三角形

给你一个数字三角形，求从顶端走到底端最大收益

$$f_{i,j} = \max(f_{i+1,j}, f_{i+1,j+1}) + a_{i,j}$$

水题

背包问题

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

多重背包问题

一个背包容量为 m , 有 n 件物品. 第 i 件物品重 w_i , 价值 v_i , 最多可以装 t_i 件. 求最大价值和.

保证 $O(m * \sum t_i)$ 的程序会被卡掉.

背包问题

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

多重背包问题

一个背包容量为 m , 有 n 件物品. 第 i 件物品重 w_i , 价值 v_i , 最多可以装 t_i 件. 求最大价值和.

保证 $O(m * \sum t_i)$ 的程序会被卡掉.

解法一: 把 t_i 二进制拆分.

解法二: 单调队列.

flappy bird

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

flappy bird

Flappy Bird 是一款风靡一时的休闲手机游戏。玩家需要不断控制点击手机屏幕的频率来调节小鸟的飞行高度，让小鸟顺利通过画面右方的管道缝隙。如果小鸟一不小心碰到了水管或者掉在地上的话，便宣告失败。为了简化问题，我们对游戏规则进行了简化和改编：

游戏界面是一个长为 n ，高为 m 的二维平面，其中有 k 个管道（忽略管道的宽度）。

小鸟始终在游戏界面内移动。小鸟从游戏界面最左边任意整数高度位置出发，到达游戏界面最右边时，游戏完成。

小鸟每个单位时间沿横坐标方向右移的距离为 1，竖直移动的距离由玩家控制。如果点击屏幕，小鸟就会上升一定高度 X ，每个单位时间可以点击多次，效果叠加；如果不点击屏幕，小鸟就会下降一定高度 Y 。小鸟位于横坐标方向不同位置时，上升的高度 X 和下降的高度 Y 可能互不相同。

小鸟高度等于 0 或者小鸟碰到管道时，游戏失败。小鸟高度为 m 时，无法再上升。

现在，请你判断是否可以完成游戏。如果可以，输出最少点击屏幕数；否则，输出小鸟最多可以通过多少个管道缝隙。

flappy bird

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

flappy bird

Flappy Bird 是一款风靡一时的休闲手机游戏。玩家需要不断控制点击手机屏幕的频率来调节小鸟的飞行高度，让小鸟顺利通过画面右方的管道缝隙。如果小鸟一不小心碰到了水管或者掉在地上的话，便宣告失败。为了简化问题，我们对游戏规则进行了简化和改编：

游戏界面是一个长为 n ，高为 m 的二维平面，其中有 k 个管道（忽略管道的宽度）。

小鸟始终在游戏界面内移动。小鸟从游戏界面最左边任意整数高度位置出发，到达游戏界面最右边时，游戏完成。

小鸟每个单位时间沿横坐标方向右移的距离为 1，竖直移动的距离由玩家控制。如果点击屏幕，小鸟就会上升一定高度 X ，每个单位时间可以点击多次，效果叠加；如果不点击屏幕，小鸟就会下降一定高度 Y 。小鸟位于横坐标方向不同位置时，上升的高度 X 和下降的高度 Y 可能互不相同。

小鸟高度等于 0 或者小鸟碰到管道时，游戏失败。小鸟高度为 m 时，无法再上升。

现在，请你判断是否可以完成游戏。如果可以，输出最少点击屏幕数；否则，输出小鸟最多可以通过多少个管道缝隙。

多重背包问题

最长上升子序列

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

最长上升子序列

给你一段长度为 k 的序列 N ，求出其最长上升子序列的长度。子序列定义如下：如果对于正整数数列 $a_1, a_2, a_3, \dots, a_p$ ，满足 $a_1 < a_2 < a_3 \cdots < a_p \leq k$ ，那么 $N_{a_1}, N_{a_2}, N_{a_3} \dots N_{a_p}$ 为序列 N 的一个子序列。

$$k \leq 100000$$

最长上升子序列

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

最长上升子序列

给你一段长度为 k 的序列 N ，求出其最长上升子序列的长度。子序列定义如下：如果对于正整数数列 $a_1, a_2, a_3, \dots, a_p$ ，满足 $a_1 < a_2 < a_3 \cdots < a_p \leq k$ ，那么 $N_{a_1}, N_{a_2}, N_{a_3} \dots N_{a_p}$ 为序列 N 的一个子序列。

$$k \leq 100000$$

$$f_i = \max(f_j + 1) \text{ where } j < i \text{ and } a_j < a_i$$

最长上升子序列

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

最长上升子序列

给你一段长度为 k 的序列 N ，求出其最长上升子序列的长度。子序列定义如下：如果对于正整数数列 $a_1, a_2, a_3, \dots, a_p$ ，满足 $a_1 < a_2 < a_3 \cdots < a_p \leq k$ ，那么 $N_{a_1}, N_{a_2}, N_{a_3} \dots N_{a_p}$ 为序列 N 的一个子序列。

$$k \leq 100000$$

$$f_i = \max(f_j + 1) \text{ where } j < i \text{ and } a_j < a_i$$

解法 1：树状数组

解法 2：单调队列

matrix

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

matrix

给你 n 个矩形，请你找到一个序列 $a_1, a_2, a_3 \dots a_k$ 满足

$a_1 < a_2 < a_3 < \dots < a_k \leq n$ ，且第 a_p 个矩形 ($1 < p < k$) 能包含（允许完全重合）第 a_{p-1} 个矩形。现按顺序给出这些矩形的边长，请输出长度最长的合法序列，如果有多个长度相同，选择 a_1 最小的，如果仍有多个合法序列，则在其中选择 a_2 最小的，以此类推。

$$n \leq 1000$$

matrix

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

matrix

给你 n 个矩形，请你找到一个序列 $a_1, a_2, a_3 \dots a_k$ 满足

$a_1 < a_2 < a_3 < \dots < a_k \leq n$ ，且第 a_p 个矩形 ($1 < p < k$) 能包含（允许完全重合）第 a_{p-1} 个矩形。现按顺序给出这些矩形的边长，请输出长度最长的合法序列，如果有多个长度相同，选择 a_1 最小的，如果仍有多个合法序列，则在其中选择 a_2 最小的，以此类推。

$$n \leq 1000$$

讲每个矩形旋转为长 $>$ 宽不影响正确性，直接用最长公共子序列做法即可，由于字典序问题，所以需要从后往前递推

没有上司的舞会

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

没有上司的舞会

Ural 大学有 N 个职员，编号为 1 到 N 。他们有从属关系，也就是说他们的关系就像一棵以校长为根的树，父结点就是子结点的直接上司。每个职员有一个快乐指数。现在有个周年庆宴会，要求与会职员的快乐指数最大。但是，没有职员愿和直接上司一起与会。

没有上司的舞会

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

没有上司的舞会

Ural 大学有 N 个职员，编号为 1 到 N 。他们有从属关系，也就是说他们的关系就像一棵以校长为根的树，父结点就是子结点的直接上司。每个职员有一个快乐指数。现在有个周年庆宴会，要求与会职员的快乐指数最大。但是，没有职员愿和直接上司一起与会。

状态很简单。

f_i 表示 i 人参加了舞会的时候这个子树的欢乐值之和。

g_i 表示 i 人没参加舞会的时候这个子树的欢乐值之和。

转移就更简单了。

$$f_i = v_i + \sum g_j | j \text{ is son of } i$$

$$g_i = \sum \max(f_j, g_j) | j \text{ is son of } i$$

Nim 游戏

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

Nim 游戏

给你一个 DAG，有一个棋子，先手后手依次将他向出边移动，最先无法移动的人输，求先手是否必胜。

Nim 游戏

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

Nim 游戏

给你一个 DAG，有一个棋子，先手后手依次将他向出边移动，最先无法移动的人输，求先手是否必胜。

为什么只会有必胜/必败？

Nim 游戏

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

Nim 游戏

给你一个 DAG，有一个棋子，先手后手依次将他向出边移动，最先无法移动的人输，求先手是否必胜。

为什么只会有必胜/必败？

只有能够转移到任意对手必败态，则为必胜态，反之必败。

硬币游戏

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

硬币游戏

最初地面上有一堆 n 个硬币 ($5 \leq n \leq 2000$), 从上面数第 i 个硬币的价值为 C_i ($1 \leq C_i \leq 100000$);

游戏开始后, A 先取一枚或两枚硬币。如果 A 取了一枚, 那么 B 可以继续取一枚或两枚; 如果 A 取了两枚, 那么 B 可以取一到四枚硬币。每次都只能从最上面取。每一次, 当前取硬币的人都至少取一枚硬币, 最多可以取他的对手上一次取硬币数目的两倍。当没有硬币可取的时候, 游戏就结束了。

然后, 他们就可以用得到的硬币向 John 买东西, 当然, 他们游戏的目的是要尽可能使自己得到的硬币价值更大。现在你的任务是, 求出在两个人都想得到更大价值的情况下, 游戏结束后, 第一个人最多能得到的硬币价值。

硬币游戏

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

硬币游戏

最初地面上有一堆 n 个硬币 ($5 \leq n \leq 2000$), 从上面数第 i 个硬币的价值为 $C_i (1 \leq C_i \leq 100000)$;

游戏开始后, A 先取一枚或两枚硬币。如果 A 取了一枚, 那么 B 可以继续取一枚或两枚; 如果 A 取了两枚, 那么 B 可以取一到四枚硬币。每次都只能从最上面取。每一次, 当前取硬币的人都至少取一枚硬币, 最多可以取他的对手上一次取硬币数目的两倍。当没有硬币可取的时候, 游戏就结束了。

然后, 他们就可以用得到的硬币向 John 买东西, 当然, 他们游戏的目的是要尽可能使自己得到的硬币价值更大。现在你的任务是, 求出在两个人都想得到更大价值的情况下, 游戏结束后, 第一个人最多能得到的硬币价值。

我能做的是转移到一个使对手最有策略收益最小的状态作为我当前状态的最大收益

Hearthstone

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

Hearthstone

在卡牌游戏《Hearthstone》中一个日经的内容就是讨论奥术飞弹击杀精灵龙的概率，这个问题属于小学奥数级别，但在喷子横行的贴吧中，Sinatra 常常发现自己完全没法说服别人，他本来想要用程序来验证的，但可惜他 OI 太弱，弄了半天都弄不出来，最后只能让你帮助一下他了。

我们将问题简要描述一下，已知奥术飞弹总共会打出 k 发，每一发都会随机选择一个存活的敌人并对其造成 1 点伤害，而场上共有 m 只精灵龙，每只均有 2 点生命，同时还有一个生命无限的敌方英雄。现在 Artanis 想问你在 k 发奥术飞弹均打出后，共计杀死了 p 只精灵龙的概率。

$$n \leq 300$$

Hearthstone

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

Hearthstone

在卡牌游戏《Hearthstone》中一个日经的内容就是讨论奥术飞弹击杀精灵龙的概率，这个问题属于小学奥数级别，但在喷子横行的贴吧中，Sinatra 常常发现自己完全没法说服别人，他本来想要用程序来验证的，但可惜他 OI 太弱，弄了半天都弄不出来，最后只能让你帮助一下他了。

我我们将问题简要描述一下，已知奥术飞弹总共会打出 k 发，每一发都会随机选择一个存活的敌人并对其造成 1 点伤害，而场上共有 m 只精灵龙，每只均有 2 点生命，同时还有一个生命无限的敌方英雄。现在 Artanis 想问你在 k 发奥术飞弹均打出后，共计杀死了 p 只精灵龙的概率。

$$n \leq 300$$

$dp_{i,j,k}$ 表示打了 i 次奥术飞弹，有 j 条龙被打掉一滴血， k 条龙被打死的概率

尼克的任务

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

尼克的任务

尼克每天上班之前都连接上英特网，接收他的上司发来的邮件，这些邮件包含了尼克主管的部门当天要完成的全部任务，每个任务由一个开始时刻与一个持续时间构成。

尼克的一个工作日为 N 分钟，从第一分钟开始到第 N 分钟结束。当尼克到达单位后他就开始干活。如果在同一时刻有多个任务需要完成，尼克可以任选其中的一个来做，而其余的则由他的同事完成，反之如果只有一个任务，则该任务必需由尼克去写成，假如某些任务开始时刻尼克正在工作，则这些任务也由尼克的同事完成。如果某任务于第 P 分钟开始，持续时间为 T 分钟，则该任务将在第 $P+T-1$ 分钟结束。

写一个程序计算尼克应该如何选取任务，才能获得最大的空暇时间。

尼克的任务

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

尼克的任务

尼克每天上班之前都连接上英特网，接收他的上司发来的邮件，这些邮件包含了尼克主管的部门当天要完成的全部任务，每个任务由一个开始时刻与一个持续时间构成。

尼克的一个工作日为 N 分钟，从第一分钟开始到第 N 分钟结束。当尼克到达单位后他就开始干活。如果在同一时刻有多个任务需要完成，尼克可以任选其中的一个来做，而其余的则由他的同事完成，反之如果只有一个任务，则该任务必需由尼克去写成，假如某些任务开始时刻尼克正在工作，则这些任务也由尼克的同事完成。如果某任务于第 P 分钟开始，持续时间为 T 分钟，则该任务将在第 $P+T-1$ 分钟结束。

写一个程序计算尼克应该如何选取任务，才能获得最大的空暇时间。

dp_i 表示从第 i 分钟开始做任务最大空闲时间
倒推 $dp[0]$

背包问题

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

- 0/1 背包问题
- 完全背包问题
- 混合背包问题
- 分组背包问题

能量项链

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

能量项链

在 Mars 星球上，每个 Mars 人都随身佩带着一串能量项链。在项链上有 N 颗能量珠。能量珠是一颗有头标记与尾标记的珠子，这些标记对应着某个正整数。并且，对于相邻的两颗珠子，前一颗珠子的尾标记一定等于后一颗珠子的头标记。因为只有这样，通过吸盘（吸盘是 Mars 人吸收能量的一种器官）的作用，这两颗珠子才能聚合成一颗珠子，同时释放出可以被吸盘吸收的能量。如果前一颗能量珠的头标记为 m ，尾标记为 r ，后一颗能量珠的头标记为 r ，尾标记为 n ，则聚合后释放的能量为（Mars 单位），新产生的珠子的头标记为 m ，尾标记为 n 。

需要时，Mars 人就用吸盘夹住相邻的两颗珠子，通过聚合得到能量，直到项链上只剩下一颗珠子为止。显然，不同的聚合顺序得到的总能量是不同的，请你设计一个聚合顺序，使一串项链释放出的总能量最大。

例如：设 $N=4$ ，4 颗珠子的头标记与尾标记依次为 $(2, 3)$ $(3, 5)$ $(5, 10)$ $(10, 2)$ 。我们用记号 \oplus 表示两颗珠子的聚合操作， $(j \oplus k)$ 表示第 j, k 两颗珠子聚合后所释放的能量。则第 4、1 两颗珠子聚合后释放的能量为：

$$(4 \oplus 1) = 10 * 2 * 3 = 60。$$

这一串项链可以得到最优值的一个聚合顺序所释放的总能量为

$$((4 \oplus 1) \oplus 2) \oplus 3 = 10 * 2 * 3 + 10 * 3 * 5 + 10 * 5 * 10 = 710$$

能量项链

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

能量项链

在 Mars 星球上，每个 Mars 人都随身佩带着一串能量项链。在项链上有 N 颗能量珠。能量珠是一颗有头标记与尾标记的珠子，这些标记对应着某个正整数。并且，对于相邻的两颗珠子，前一颗珠子的尾标记一定等于后一颗珠子的头标记。因为只有这样，通过吸盘（吸盘是 Mars 人吸收能量的一种器官）的作用，这两颗珠子才能聚合成一颗珠子，同时释放出可以被吸盘吸收的能量。如果前一颗能量珠的头标记为 m ，尾标记为 r ，后一颗能量珠的头标记为 r ，尾标记为 n ，则聚合后释放的能量为 (Mars 单位) ，新产生的珠子的头标记为 m ，尾标记为 n 。

需要时，Mars 人就用吸盘夹住相邻的两颗珠子，通过聚合得到能量，直到项链上只剩下一颗珠子为止。显然，不同的聚合顺序得到的总能量是不同的，请你设计一个聚合顺序，使一串项链释放出的总能量最大。

例如：设 $N=4$ ，4 颗珠子的头标记与尾标记依次为 $(2, 3)$ $(3, 5)$ $(5, 10)$ $(10, 2)$ 。我们用记号 \oplus 表示两颗珠子的聚合操作， $(j \oplus k)$ 表示第 j, k 两颗珠子聚合后所释放的能量。则第 4、1 两颗珠子聚合后释放的能量为：

$$(4 \oplus 1) = 10 * 2 * 3 = 60。$$

这一串项链可以得到最优值的一个聚合顺序所释放的总能量为

$$((4 \oplus 1) \oplus 2) \oplus 3 = 10 * 2 * 3 + 10 * 3 * 5 + 10 * 5 * 10 = 710$$

$dp_{i,j}$ 表示讲第 i 个珠子到第 j 个珠子合并的最小代价

滑雪

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

滑雪

trs 喜欢滑雪。他来到了一个滑雪场，这个滑雪场是一个矩形，为了简便，我们用 r 行 c 列的矩阵来表示每块地形。为了得到更快的速度，滑行的路线必须向下倾斜。

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

例如样例中的那个矩形，可以从某个点滑向上下左右四个相邻的点之一。

例如 $24-17-16-1$ ，可以看出 $25-24-23\dots 3-2-1$ 是最长的一条。

滑雪

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

滑雪

trs 喜欢滑雪。他来到了一个滑雪场，这个滑雪场是一个矩形，为了简便，我们用 r 行 c 列的矩阵来表示每块地形。为了得到更快的速度，滑行的路线必须向下倾斜。

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

例如样例中的那个矩形，可以从某个点滑向上下左右四个相邻的点之一。

例如 $24-17-16-1$ ，可以看出 $25-24-23\dots 3-2-1$ 是最长的一条。

$dp_{i,j}$ 表示从 (i,j) 位置向下滑最远距离，通过搜索转移

EasyNum

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

EasyNum

我们定义一个数的权值 P 为这个数的各位数字之和，给定一个区间 $[L,R]$ 和一个整数 K ，请输出在区间 $[L,R]$ 中且权值为 K 的数的个数。

EasyNum

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

EasyNum

我们定义一个数的权值 P 为这个数的各位数字之和，给定一个区间 $[L,R]$ 和一个整数 K ，请输出在区间 $[L,R]$ 中且权值为 K 的数的个数。

EasyNum

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

```
typedef long long qword;
qword f[20][800][2]; // f[i][j][over] 表示处理到第 i 位，和为 j，是否顶上限的方案个数
short num[20]; // 当前需要处理的上界
int n, m; // n: 位数 m: 和
qword dp(int now, int sum, bool over)
{
    if (~f[now][sum][over]) return f[now][sum][over];
    qword ret = 0;
    if (now == n) return sum == m;
    if (sum > m) return 0;
    if (over)
    {
        for (int i = 0; i < 10; i++)
            ret += dp(now + 1, sum + i, 1);
    } else
    {
        for (int i = 0; i < num[now]; i++)
            ret += dp(now + 1, sum + i, 1);
        ret += dp(now + 1, sum + num[now], 0);
    }
    return f[now][sum][over] = ret;
}

qword work(qword v)
{
    if (v <= -1) return 0;
    // TODO: generate array num from v
    memset(f, -1, sizeof(f));
    return dp(0, 0, 0);
}
```

Windy 数

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

Windy 数

windy 定义了一种 windy 数。

不含前导零且相邻两个数字之差至少为 2 的正整数被称为 windy 数。

windy 想知道, 在 A 和 B 之间, 包括 A 和 B, 总共有多少个 windy 数?

Windy 数

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

Windy 数

windy 定义了一种 windy 数。

不含前导零且相邻两个数字之差至少为 2 的正整数被称为 windy 数。

windy 想知道, 在 A 和 B 之间, 包括 A 和 B, 总共有多少个 windy 数?

$dp_{i,last,over}$ 表示处理到第 i 位, 上一位位 j , 是否顶上界的方案总数

Seven

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

Seven

据说 7 是一个十分神奇的数，它拥有至少 3 个与 7 有关的性质，例如它是最小的存在某位为 7 的正整数，最小的各位数字之和为 7 的倍数的正整数，最小的为 7 的倍数的正整数，所以我们发现 7 和 7 很有关系，就是说 7 和 7 是相关的，同样的，我们可以定义一个数与 7 无关当且仅当：

- 这个数任何一位均非 7。
- 这个数各位数字之和不是 7 的倍数。
- 这个数本身不是 7 的倍数。

现在给出一个区间 $[L,R]$ ，试求出所有在 $[L,R]$ 内与 7 无关的数的平方和。

Seven

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

Seven

据说 7 是一个十分神奇的数，它拥有至少 3 个与 7 有关的性质，例如它是最小的存在某位为 7 的正整数，最小的各位数字之和为 7 的倍数的正整数，最小的为 7 的倍数的正整数，所以我们发现 7 和 7 很有关系，就是说 7 和 7 是相关的，同样的，我们可以定义一个数与 7 无关当且仅当：

- 这个数任何一位均非 7。
- 这个数各位数字之和不是 7 的倍数。
- 这个数本身不是 7 的倍数。

现在给出一个区间 $[L,R]$ ，试求出所有在 $[L,R]$ 内与 7 无关的数的平方和。

$$(10a + b)^2 = 100a^2 + 20ab + b^2$$

在数位 dp 的基础上分别记录“和”和“平方和”即可

炮兵阵地

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

炮兵阵地

司令部的将军们打算在 $N \times M$ 的网格地图上部署他们的炮兵部队。一个 $N \times M$ 的地图由 N 行 M 列组成，地图的每一格可能是山地（用“H”表示），也可能是平原（用“P”表示），如下图。在每一格平原地形上最多可以布置一支炮兵部队（山地上不能够部署炮兵部队）；一支炮兵部队在地图上的攻击范围如图中黑色区域所示：

P	P	H	P	H	H	P	P
P	H	P	H	P	H	P	P
P	P	P	H	H	H	P	H
H	P	H	P	P	P	P	H
H	P	P	P	P	H	P	H
H	P	P	H	P	H	H	P
H	H	H	P	P	P	P	H

如果在地图中的灰色所标识的平原上部署一支炮兵部队，则图中的黑色的网格表示它能够攻击到的区域：沿横向左右各两格，沿纵向上下各两格。图上其它白色网格均攻击不到。从图上可见炮兵的攻击范围不受地形的影响。

现在，将军们规划如何部署炮兵部队，在防止误伤的前提下（保证任何两支炮兵部队之间不能互相攻击，即任何一支炮兵部队都不在其他支炮兵部队的攻击范围内），在整个地图区域内最多能够摆放多少我军的炮兵部队。

中国象棋

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

中国象棋

在 $n * m$ 的中国象棋棋盘上放若干个炮, 使它们互相不能攻击. 求方案数.
 $n, m \leq 100$.

中国象棋

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

中国象棋

在 $n * m$ 的中国象棋棋盘上放若干个炮, 使它们互相不能攻击. 求方案数.
 $n, m \leq 100$.

每行每列都至多有 2 个炮.

如果一行一行地考虑, 那么我们只需要知道已经放了 2 个炮, 1 个炮, 没放炮的列各有多少就行了.

$f_{i,j,k}$ 表示考虑了 i 行, 有 j 列有 0 个炮, k 列有 1 个炮的方案数.

转移 easy.

传纸条

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

传纸条

小渊和小轩是好朋友也是同班同学，他们在一起总有谈不完的话题。一次素质拓展活动中，班上同学安排做成一个 m 行 n 列的矩阵，而小渊和小轩被安排在矩阵对角线的两端，因此，他们就无法直接交谈了。幸运的是，他们可以通过传纸条来进行交流。纸条要经由许多同学传到对方手里，小渊坐在矩阵的左上角，坐标 $(1,1)$ ，小轩坐在矩阵的右下角，坐标 (m,n) 。从小渊传到小轩的纸条只可以向下或者向右传递，从小轩传给小渊的纸条只可以向上或者向左传递。

在活动进行中，小渊希望给小轩传递一张纸条，同时希望小轩给他回复。班里每个同学都可以帮他们传递，但只会帮他们一次，也就是说如果此人在小渊递给小轩纸条的时候帮忙，那么在小轩递给小渊的时候就不会再帮忙。反之亦然。

还有一件事情需要注意，全班每个同学愿意帮忙的好感度有高有低（注意：小渊和小轩的好心程度没有定义，输入时用 0 表示），可以用一个 0-100 的自然数来表示，数越大表示越好心。小渊和小轩希望尽可能找好心程度高的同学来帮忙传纸条，即找到来回两条传递路径，使得这两条路径上同学的好心程度只和最大。现在，请你帮助小渊和小轩找到这样的两条路径。

传纸条

mhy12345

其他

传纸条

小渊和小轩是好朋友也是同班同学，他们在一起总有谈不完的话题。一次素质拓展活动中，班上同学安排做成一个 m 行 n 列的矩阵，而小渊和小轩被安排在矩阵对角线的两端，因此，他们就无法直接交谈了。幸运的是，他们可以通过传纸条来进行交流。纸条要经由许多同学传到对方手里，小渊坐在矩阵的左上角，坐标 $(1,1)$ ，小轩坐在矩阵的右下角，坐标 (m,n) 。从小渊传到小轩的纸条只可以向下或者向右传递，从小轩传给小渊的纸条只可以向上或者向左传递。

在活动进行中，小渊希望给小轩传递一张纸条，同时希望小轩给他回复。班里每个同学都可以帮他们传递，但只会帮他们一次，也就是说如果此人在小渊递给小轩纸条的时候帮忙，那么在小轩递给小渊的时候就不会再帮忙。反之亦然。

还有一件事情需要注意，全班每个同学愿意帮忙的好感度有高有低（注意：小渊和小轩的好心程度没有定义，输入时用 0 表示），可以用一个 0-100 的自然数来表示，数越大表示越好心。小渊和小轩希望尽可能找好心程度高的同学来帮忙传纸条，即找到来回两条传递路径，使得这两条路径上同学的好心程度之和最大。现在，请你帮助小渊和小轩找到这样的两条路径。

看做两个纸条分别从左上角同时向右下角传.

$dp_{i,j,k}$ 表示第 i 秒两个纸条横坐标依次为 j,k , 最大满意度

矩阵乘法

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

矩阵乘法

$$A = B \times C$$

$$A_{i,j} = \sum B_{i,k} * C_{k,j}$$

矩阵乘法

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

矩阵乘法

$$A = B \times C$$

$$A_{i,j} = \sum B_{i,k} * C_{k,j}$$

```
struct matrix
{
    static const int maxn = 100;
    int n,m;
    int a[maxn][maxn];
};

void multiply(matrix &res,const matrix &m1,const matrix &m2)
{
    assert(m1.m == m2.n);
    memset(res.a,0,sizeof(res.a));
    for (int i=0;i<m1.n;i++)
        for (int j=0;j<m2.m;j++)
            for (int k=0;k<m1.m;k++)
                res.a[i][j]+=m1.a[i][k]*m2.a[k][j];
}
```

斐波那契数列

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

斐波那契数列

求模 P 意义下斐波那契数列第 K 项

$$K \leq 10^{18}$$

斐波那契数列

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

斐波那契数列

求模 P 意义下斐波那契数列第 K 项

$$K \leq 10^{18}$$

$$f_{i,0} = f_{i-1,1}$$

$$f_{i,1} = f_{i-1,0} + f_{i-1,1}$$

$$T = \begin{Bmatrix} 0 & 1 \\ 1 & 1 \end{Bmatrix}$$

矩阵乘法优化 dp

矩阵游戏

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

矩阵游戏

婷婷是个喜欢矩阵的小朋友，有一天她想用电脑生成一个巨大的 n 行 m 列的矩阵（你不用担心她如何存储）。她生成的这个矩阵满足一个神奇的性质：若用 $F[i][j]$ 来表示矩阵中第 i 行第 j 列的元素，则 $F[i][j]$ 满足下面的递推式：

$$F[1][1] = 1$$

$$F[i][j] = a * F[i][j-1] + b(j! = 1)$$

$$F[i][1] = c * F[i-1][m] + d(i! = 1)$$

递推式中 a, b, c, d 都是给定的常数。

现在婷婷想知道 $F[n][m]$ 的值是多少，请你帮助她。由于最终结果可能很大，你只需要输出 $F[n][m]$ 除以 1,000,000,007 的余数。

矩阵游戏

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

矩阵游戏

婷婷是个喜欢矩阵的小朋友，有一天她想用电脑生成一个巨大的 n 行 m 列的矩阵 (你不用担心她如何存储)。她生成的这个矩阵满足一个神奇的性质：若用 $F[i][j]$ 来表示矩阵中第 i 行第 j 列的元素，则 $F[i][j]$ 满足下面的递推式：

$$F[1][1] = 1$$

$$F[i][j] = a * F[i][j-1] + b(j! = 1)$$

$$F[i][1] = c * F[i-1][m] + d(i! = 1)$$

递推式中 a, b, c, d 都是给定的常数。

现在婷婷想知道 $F[n][m]$ 的值是多少，请你帮助她。由于最终结果可能很大，你只需要输出 $F[n][m]$ 除以 1,000,000,007 的余数。

先通过矩阵乘法算出行转移矩阵，然后在用矩阵乘法计算列，十进制快速幂优化常数

黑暗星球之战 II——反制

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

黑暗星球之战 II——反制

“En Taro Dark! 我们已经知晓了 Light 联盟的阴谋, 现在让我们来展示一下黑暗算法的巨大威力!”——A0A0

A0A0 通过黑暗星球的情报手段提早得到了 Light 联盟的计划, 他准备使用超级反制器来进行反击。但他所不知道的是, 在上一次机房星区对 A0A0 的突袭的最后一步, 就是把 A0A0 抬起来耍的时候, Light 联盟的特工已经趁乱修改了超级反制器的设置。当 A0A0 发现这一点的时候, 已经来不及再去修改了, 只能强行启动超级反制器了。

现在的超级反制器有 M 个能量槽, 每一轮中会随机选出一个并充入 1 单位的能量, 共执行 T 轮, 但由于上一次的破坏, 当一个能量槽的能量达到 N 的时候这个槽中的能量将会清零。启动前每个能量槽都有一定初始能量 (保证小于 N), 现在 A0A0 想知道在运行结束后期望意义下每个能量值的能量槽的个数。作为一个双面间谍, 这个任务又再一次交给你了。

黑暗星球之战 II——反制

动态规划

mhy12345

转移运算

sum

max/min

博弈

概率

状态特点分类

线性 dp

背包 dp

区间 dp

记忆化搜索

数位 dp

状压 dp

其他

矩阵乘法

黑暗星球之战 II——反制

“En Taro Dark! 我们已经知晓了 Light 联盟的阴谋, 现在让我们来展示一下黑暗算法的巨大威力!”——A0A0

A0A0 通过黑暗星球的情报手段提早得到了 Light 联盟的计划, 他准备使用超级反制器来进行反击。但他所不知道的是, 在上一次机房星区对 A0A0 的突袭的最后一步, 就是把 A0A0 抬起来耍的时候, Light 联盟的特工已经趁乱修改了超级反制器的设置。当 A0A0 发现这一点的时候, 已经来不及再去修改了, 只能强行启动超级反制器了。

现在的超级反制器有 M 个能量槽, 每一轮中会随机选出一个并充入 1 单位的能量, 共执行 T 轮, 但由于上一次的破坏, 当一个能量槽的能量达到 N 的时候这个槽中的能量将会清零。启动前每个能量槽都有一定初始能量 (保证小于 N), 现在 A0A0 想知道在运行结束后期望意义下每个能量值的能量槽的个数。作为一个双面间谍, 这个任务又再一次交给了你。

dp_i 表示原来才 pos 的位置的能量执行 2^t 轮后在 $pos+i$ 位置的概率
倍增处理