

动态规划进阶

Claris

Hangzhou Dianzi University

2018 年 2 月 9 日

Overview

- 本节课介绍几种动态规划优化的方法：

Overview

- 本节课介绍几种动态规划优化的方法：
- 调换状态与值。

Overview

- 本节课介绍几种动态规划优化的方法：
- 调换状态与值。
- 细化状态转移。

Overview

- 本节课介绍几种动态规划优化的方法：
- 调换状态与值。
- 细化状态转移。
- 简化状态数。

调换状态与值

- 顾名思义，就是调换状态与 DP 值。

调换状态与值

- 顾名思义，就是调换状态与 DP 值。
- 比如经典 01 背包中状态是 f_i 表示容量 i 最多背走多少价值的物品。

调换状态与值

- 顾名思义，就是调换状态与 DP 值。
- 比如经典 01 背包中状态是 f_i 表示容量 i 最多背走多少价值的物品。
- 调换后就是： f_i 表示背走价值为 i 的物品至少需要多少容量。

调换状态与值

- 顾名思义，就是调换状态与 DP 值。
- 比如经典 01 背包中状态是 f_i 表示容量 i 最多背走多少价值的物品。
- 调换后就是： f_i 表示背走价值为 i 的物品至少需要多少容量。
- 有时候调换状态与 DP 值可以有意想不到的效果。

最长公共子序列

一个串的子序列是指去掉某些位置后剩下的部分，不一定连续。

最长公共子序列

一个串的子序列是指去掉某些位置后剩下的部分，不一定连续。

给定两个小写字母串 S 和 T ，请求出它们最长公共子序列的长度。

最长公共子序列

一个串的子序列是指去掉某些位置后剩下的部分，不一定连续。

给定两个小写字母串 S 和 T ，请求出它们最长公共子序列的长度。

- $|S| \leq 3000$ 。

最长公共子序列

一个串的子序列是指去掉某些位置后剩下的部分，不一定连续。

给定两个小写字母串 S 和 T ，请求出它们最长公共子序列的长度。

- $|S| \leq 3000$ 。
- $|T| \leq 500000$ 。

Naive Solution

- 考虑常规做法。

Naive Solution

- 考虑常规做法。
- 状态： $f_{i,j}$ 表示 $S[1,i]$ 与 $T[1,j]$ 的 LCS。

Naive Solution

- 考虑常规做法。
- 状态： $f_{i,j}$ 表示 $S[1,i]$ 与 $T[1,j]$ 的 LCS。
- 若 $S_i = T_j$ ，则 $f_{i,j} = f_{i-1,j-1} + 1$ 。

Naive Solution

- 考虑常规做法。
- 状态： $f_{i,j}$ 表示 $S[1,i]$ 与 $T[1,j]$ 的 LCS。
- 若 $S_i = T_j$ ，则 $f_{i,j} = f_{i-1,j-1} + 1$ 。
- 否则 $f_{i,j} = \max(f_{i,j-1}, f_{i-1,j})$ 。

Naive Solution

- 考虑常规做法。
- 状态： $f_{i,j}$ 表示 $S[1,i]$ 与 $T[1,j]$ 的 LCS。
- 若 $S_i = T_j$ ，则 $f_{i,j} = f_{i-1,j-1} + 1$ 。
- 否则 $f_{i,j} = \max(f_{i,j-1}, f_{i-1,j})$ 。
- $ans = f_{|S|,|T|}$ 。

Naive Solution

- 考虑常规做法。
- 状态： $f_{i,j}$ 表示 $S[1,i]$ 与 $T[1,j]$ 的 LCS。
- 若 $S_i = T_j$ ，则 $f_{i,j} = f_{i-1,j-1} + 1$ 。
- 否则 $f_{i,j} = \max(f_{i,j-1}, f_{i-1,j})$ 。
- $ans = f_{|S|,|T|}$ 。
- 时间复杂度 $O(|S||T|)$ 。

细化状态转移

- 有时候 DP 复杂度太大是因为状态转移复杂度过高。

细化状态转移

- 有时候 DP 复杂度太大是因为状态转移复杂度过高。
- 可以考虑将转移的过程再次分解、细化。

细化状态转移

- 有时候 DP 复杂度太大是因为状态转移复杂度过高。
- 可以考虑将转移的过程再次分解、细化。
- 比如三个人同时各走一步，可以分解成 A 先走一步， B 再走一步， C 再走一步。

细化状态转移

- 有时候 DP 复杂度太大是因为状态转移复杂度过高。
- 可以考虑将转移的过程再次分解、细化。
- 比如三个人同时各走一步，可以分解成 A 先走一步， B 再走一步， C 再走一步。
- 这样转移的复杂度就从三次方降低到了一次方。

Wavel Sequence

一个序列 a_1, a_2, \dots, a_n 是“波浪的”，当且仅当

$$a_1 < a_2 > a_3 < a_4 > a_5 < a_6 \dots。$$

Wavel Sequence

一个序列 a_1, a_2, \dots, a_n 是“波浪的”，当且仅当

$$a_1 < a_2 > a_3 < a_4 > a_5 < a_6 \dots$$

给定序列 a_1, a_2, \dots, a_n 和 b_1, b_2, \dots, b_m ，小 Q 希望找到两个序列 $f_1, f_2, \dots, f_k (1 \leq f_i \leq n, f_i < f_{i+1})$ 和 $g_1, g_2, \dots, g_k (1 \leq g_i \leq m, g_i < g_{i+1})$ ，满足 $a_{f_i} = b_{g_i}$ 恒成立且序列 $a_{f_1}, a_{f_2}, \dots, a_{f_k}$ 是“波浪的”。

Wavel Sequence

一个序列 a_1, a_2, \dots, a_n 是“波浪的”，当且仅当

$$a_1 < a_2 > a_3 < a_4 > a_5 < a_6 \dots$$

给定序列 a_1, a_2, \dots, a_n 和 b_1, b_2, \dots, b_m ，小 Q 希望找到两个序列 $f_1, f_2, \dots, f_k (1 \leq f_i \leq n, f_i < f_{i+1})$ 和 $g_1, g_2, \dots, g_k (1 \leq g_i \leq m, g_i < g_{i+1})$ ，满足 $a_{f_i} = b_{g_i}$ 恒成立且序列 $a_{f_1}, a_{f_2}, \dots, a_{f_k}$ 是“波浪的”。

求有多少对 f 和 g 满足条件。

Wavel Sequence

一个序列 a_1, a_2, \dots, a_n 是“波浪的”，当且仅当

$$a_1 < a_2 > a_3 < a_4 > a_5 < a_6 \dots$$

给定序列 a_1, a_2, \dots, a_n 和 b_1, b_2, \dots, b_m ，小 Q 希望找到两个序列 $f_1, f_2, \dots, f_k (1 \leq f_i \leq n, f_i < f_{i+1})$ 和 $g_1, g_2, \dots, g_k (1 \leq g_i \leq m, g_i < g_{i+1})$ ，满足 $a_{f_i} = b_{g_i}$ 恒成立且序列 $a_{f_1}, a_{f_2}, \dots, a_{f_k}$ 是“波浪的”。

求有多少对 f 和 g 满足条件。

- $n, m, a_i, b_i \leq 2000$ 。

Wavel Sequence

一个序列 a_1, a_2, \dots, a_n 是 “波浪的” , 当且仅当

$$a_1 < a_2 > a_3 < a_4 > a_5 < a_6 \dots$$

给定序列 a_1, a_2, \dots, a_n 和 b_1, b_2, \dots, b_m , 小 Q 希望找到两个序列 $f_1, f_2, \dots, f_k (1 \leq f_i \leq n, f_i < f_{i+1})$ 和 $g_1, g_2, \dots, g_k (1 \leq g_i \leq m, g_i < g_{i+1})$, 满足 $a_{f_i} = b_{g_i}$ 恒成立且序列 $a_{f_1}, a_{f_2}, \dots, a_{f_k}$ 是 “波浪的”。

求有多少对 f 和 g 满足条件。

■ $n, m, a_i, b_i \leq 2000$ 。

■ Source : 2017 Multi-University Training Contest 4

Solution

- 设 $f_{i,j,k}$ 表示仅考虑 $a[1, i]$ 与 $b[1, j]$, 选择的两个子序列结尾分别是 a_i 和 b_j , 且上升下降状态是 k 时的方案数。

Solution

- 设 $f_{i,j,k}$ 表示仅考虑 $a[1,i]$ 与 $b[1,j]$, 选择的两个子序列结尾分别是 a_i 和 b_j , 且上升下降状态是 k 时的方案数。
- $f_{i,j,k} = \sum f_{x,y,1-k}$, 其中 $x < i, y < j$ 。

Solution

- 设 $f_{i,j,k}$ 表示仅考虑 $a[1, i]$ 与 $b[1, j]$, 选择的两个子序列结尾分别是 a_i 和 b_j , 且上升下降状态是 k 时的方案数。
- $f_{i,j,k} = \sum f_{x,y,1-k}$, 其中 $x < i, y < j$ 。
- 暴力转移的时间复杂度为 $O(n^4)$, 不能接受。

简化状态数

- 顾名思义，就是简化状态数。

简化状态数

- 顾名思义，就是简化状态数。
- 需要深入分析问题的性质，来简化状态数。

整数拆分

令 $f(n)$ 表示将 n 进行分拆的方案数。

整数拆分

令 $f(n)$ 表示将 n 进行分拆的方案数。

例如, $f(4) = 1 + 1 + 1 + 1 = 1 + 1 + 2 = 1 + 3 = 2 + 2 = 4$,

所以 $f(4) = 5$ 。

整数拆分

令 $f(n)$ 表示将 n 进行分拆的方案数。

例如, $f(4) = 1 + 1 + 1 + 1 = 1 + 1 + 2 = 1 + 3 = 2 + 2 = 4$,

所以 $f(4) = 5$ 。

给定 n , 求 $f(1), f(2), \dots, f(n)$ 。

整数拆分

令 $f(n)$ 表示将 n 进行分拆的方案数。

例如, $f(4) = 1 + 1 + 1 + 1 = 1 + 1 + 2 = 1 + 3 = 2 + 2 = 4$,

所以 $f(4) = 5$ 。

给定 n , 求 $f(1), f(2), \dots, f(n)$ 。

■ $n \leq 100000$ 。

Solution

- 即求完全背包方案数。

Solution

- 即求完全背包方案数。
- 设 $f_{i,j}$ 表示 $[1, i]$ 和为 j 的方案数。

Solution

- 即求完全背包方案数。
- 设 $f_{i,j}$ 表示 $[1, i]$ 和为 j 的方案数。
- 转移：
 - 要么停止加入 $i : f_{i,j} \rightarrow f_{i+1,j}$ 。
 - 要么继续加入 $i : f_{i,j} \rightarrow f_{i,j+i}$ 。

Solution

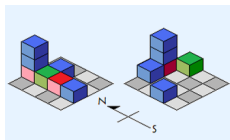
- 即求完全背包方案数。
- 设 $f_{i,j}$ 表示 $[1, i]$ 和为 j 的方案数。
- 转移：
 - 要么停止加入 $i : f_{i,j} \rightarrow f_{i+1,j}$ 。
 - 要么继续加入 $i : f_{i,j} \rightarrow f_{i,j+i}$ 。
- 状态数 $O(n^2)$ ，不能接受。

三维积木

积木是一些 $1 \times 1 \times 1$ 的小方块，每个方块都有一种颜色，红色，绿色或蓝色。玩积木的场所是一个 $N \times N$ 的网格。

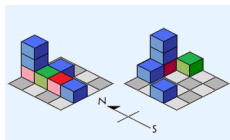
三维积木

积木是一些 $1 \times 1 \times 1$ 的小方块，每个方块都有一种颜色，红色，绿色或蓝色。玩积木的场所是一个 $N \times N$ 的网格。



三维积木

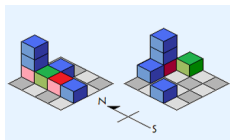
积木是一些 $1 \times 1 \times 1$ 的小方块，每个方块都有一种颜色，红色，绿色或蓝色。玩积木的场所是一个 $N \times N$ 的网格。



为了优美，积木堆从正前方看过去时都只能看到一个颜色。

三维积木

积木是一些 $1 \times 1 \times 1$ 的小方块，每个方块都有一种颜色，红色，绿色或蓝色。玩积木的场所是一个 $N \times N$ 的网格。

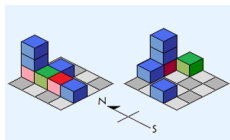


为了优美，积木堆从正前方看过去时都只能看到一个颜色。

已知每种颜色的积木个数，要求全部用上。求有多少种不同的优美的积木组合。颜色相同的积木被视为是完全相同的。

三维积木

积木是一些 $1 \times 1 \times 1$ 的小方块，每个方块都有一种颜色，红色，绿色或蓝色。玩积木的场所是一个 $N \times N$ 的网格。



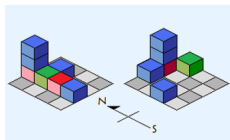
为了优美，积木堆从正前方看过去时都只能看到一个颜色。

已知每种颜色的积木个数，要求全部用上。求有多少种不同的优美的积木组合。颜色相同的积木被视为是完全相同的。

- $R, G, B, N \leq 25$ 。

三维积木

积木是一些 $1 \times 1 \times 1$ 的小方块，每个方块都有一种颜色，红色，绿色或蓝色。玩积木的场所是一个 $N \times N$ 的网格。



为了优美，积木堆从正前方看过去时都只能看到一个颜色。

已知每种颜色的积木个数，要求全部用上。求有多少种不同的优美的积木组合。颜色相同的积木被视为是完全相同的。

■ $R, G, B, N \leq 25$ 。

■ Source : BZOJ 4313

Solution

- 不妨设 R 是唯一可以看到的颜色。

Solution

- 不妨设 R 是唯一可以看到的颜色。
- 设 $f_{i,j,k,t,x,y,z}$ 表示考虑到了 (i,j) , (i,j) 的高度是 k , 第 i 行最高高度是 t , 已经用了 x 个 R , y 个 G , z 个 B 的方案数。

Solution

- 不妨设 R 是唯一可以看到的颜色。
- 设 $f_{i,j,k,t,x,y,z}$ 表示考虑到了 (i,j) , (i,j) 的高度是 k , 第 i 行最高高度是 t , 已经用了 x 个 R , y 个 G , z 个 B 的方案数。
- 转移则是要么使自己的高度 $+1$, 要么考虑下一个位置。

Solution

- 不妨设 R 是唯一可以看到的颜色。
- 设 $f_{i,j,k,t,x,y,z}$ 表示考虑到了 (i,j) , (i,j) 的高度是 k , 第 i 行最高高度是 t , 已经用了 x 个 R , y 个 G , z 个 B 的方案数。
- 转移则是要么使自己的高度 $+1$, 要么考虑下一个位置。
- 状态数 $O(n^7)$, 不能接受。

gems gems gems

n 颗钻石排成一行，价值分别为 V_1, V_2, \dots, V_n 。

gems gems gems

n 颗钻石排成一行，价值分别为 V_1, V_2, \dots, V_n 。

Alice 和 Bob 轮流操作，Alice 先手，一开始 Alice 需要选择从左侧拿走 1 到 2 颗钻石。

gems gems gems

n 颗钻石排成一行，价值分别为 V_1, V_2, \dots, V_n 。

Alice 和 Bob 轮流操作，Alice 先手，一开始 Alice 需要选择从左侧拿走 1 到 2 颗钻石。

每次若对方上一次拿了 k 个钻石，则当前方必须从左侧拿走 k 或 $k+1$ 颗钻石。

gems gems gems

n 颗钻石排成一行，价值分别为 V_1, V_2, \dots, V_n 。

Alice 和 Bob 轮流操作，Alice 先手，一开始 Alice 需要选择从左侧拿走 1 到 2 颗钻石。

每次若对方上一次拿了 k 个钻石，则当前方必须从左侧拿走 k 或 $k+1$ 颗钻石。

游戏结束当且仅当不能行动或者钻石取尽。

gems gems gems

n 颗钻石排成一行，价值分别为 V_1, V_2, \dots, V_n 。

Alice 和 Bob 轮流操作，Alice 先手，一开始 Alice 需要选择从左侧拿走 1 到 2 颗钻石。

每次若对方上一次拿了 k 个钻石，则当前方必须从左侧拿走 k 或 $k + 1$ 颗钻石。

游戏结束当且仅当不能行动或者钻石取尽。

双方都以最优策略进行游戏，即最大化自己拿到的钻石的总价值减去对方拿到的钻石的总价值。

gems gems gems

n 颗钻石排成一行，价值分别为 V_1, V_2, \dots, V_n 。

Alice 和 Bob 轮流操作，Alice 先手，一开始 Alice 需要选择从左侧拿走 1 到 2 颗钻石。

每次若对方上一次拿了 k 个钻石，则当前方必须从左侧拿走 k 或 $k + 1$ 颗钻石。

游戏结束当且仅当不能行动或者钻石取尽。

双方都以最优策略进行游戏，即最大化自己拿到的钻石的总价值减去对方拿到的钻石的总价值。

求最后双方拿走的钻石的总价值的差值。

gems gems gems

n 颗钻石排成一行，价值分别为 V_1, V_2, \dots, V_n 。

Alice 和 Bob 轮流操作，Alice 先手，一开始 Alice 需要选择从左侧拿走 1 到 2 颗钻石。

每次若对方上一次拿了 k 个钻石，则当前方必须从左侧拿走 k 或 $k + 1$ 颗钻石。

游戏结束当且仅当不能行动或者钻石取尽。

双方都以最优策略进行游戏，即最大化自己拿到的钻石的总价值减去对方拿到的钻石的总价值。

求最后双方拿走的钻石的总价值的差值。

- $n \leq 20000$ 。

gems gems gems

n 颗钻石排成一行，价值分别为 V_1, V_2, \dots, V_n 。

Alice 和 Bob 轮流操作，Alice 先手，一开始 Alice 需要选择从左侧拿走 1 到 2 颗钻石。

每次若对方上一次拿了 k 个钻石，则当前方必须从左侧拿走 k 或 $k+1$ 颗钻石。

游戏结束当且仅当不能行动或者钻石取尽。

双方都以最优策略进行游戏，即最大化自己拿到的钻石的总价值减去对方拿到的钻石的总价值。

求最后双方拿走的钻石的总价值的差值。

■ $n \leq 20000$ 。

■ Source : 2017 ACM/ICPC 沈阳站网络赛

题目提交

课上例题：

http://acm.hdu.edu.cn/diy/contest_show.php?cid=33140

课后习题：

http://acm.hdu.edu.cn/diy/contest_show.php?cid=33141

密码：

G*&GSF&*t387tr

Thank you!