线性动态规划入门与背包问题

wrpwrp

课前提示

- 上课的时候专心听讲解，**不要跟着老师抄代码**，下课后独立完成。
- 不使用 AI 做题，AI 会做不等于自己会。
- 不抄袭题解（含对照题解抄一遍），抄对不等于会做。

  看完题解后，关闭题解独立练习。

  练习中途遇到问题，应当分析题目及自己的思路，而非回忆题解或再次参考题解。
- 做过的题在课后需要重新独立完成，不参考老师的课件、代码，不参考自己以前的代码。

前言

今天的主要内容：复习序列上的 dp 和背包问题。

动态规划的核心要素

状态定义

用一个变量/数组记录子问题的最优解

状态转移方程

子问题之间的关系（如何由小问题推到大问题）

初始条件（边界）

最简单的子问题的答案

线性 DP

线性 DP

特点：

状态的计算只依赖于前面若干个状态，像一条线一样推进

换句话说，状态总是定义成类似“ $f[i]$ 表示考虑序列的前 i 个数的答案”。

P10250 [GESP样题 六级] 下楼梯

题目描述

复制 Markdown 展开 进入 IDE 模式

顽皮的小明发现，下楼梯时每步可以走 1 个台阶、2 个台阶或 3 个台阶。现在一共有 N 个台阶，你能帮小明算算有多少种方案吗?

输入格式

输入一行，包含一个整数 N。

输出格式

输出一行一个整数表示答案。

输入输出样例

输入 #1 复制

```
4
```

输出 #1 复制

```
7
```

输入 #2 复制

```
10
```

输出 #2 复制

```
274
```

说明/提示

对全部的测试点，保证 $1 \leq N \leq 60$。

P10250 [GESP样题 六级] 下楼梯

可以发现， 我们把楼梯看成序列， 我们其实就是在序列上走。每次可以走一步或者两步或者三步， 求不同的走法个数。

考虑线性 DP， 把“到第 i 级的走法数”记作 $f[i]$。

如何转移？

P10250 [GESP样题 六级] 下楼梯

我们知道， DP 是一个 “状态转移” 的过程， 我们考虑哪些状态会转移到当前状态。

在这题中， 也就是哪些位置走一步会走到当前我们的位置 i， 于是我们可以考虑最后一步走了多远。

P10250 [GESP样题 六级] 下楼梯

要到第 i 级，最后一步可能是：

从 $i-1$ 走 1 级过来（有 $f[i-1]$ 种）

从 $i-2$ 走 2 级过来（有 $f[i-2]$ 种）

从 $i-3$ 走 3 级过来 （有 $f[i-3]$ 种）

状态转移方程：

$$f[i] = f[i-1] + f[i-2] + f[i-3]$$

初值：

$f[0] = 1$：站在地面不动算一种“空走法”，是递推的基石。

$f[1] = 1$：只有 [1]。

$f[2] = 2$：[1+1]、[2]。

B3637 最长上升子序列 $\boxed{a_j}$ a_i

1 2 3 4　1 2 2 3

题目描述

复制 Markdown　展开　进入 IDE 模式

$\boxed{a_j} < a_i$

这是一个简单的动规板子题。

给出一个由 $n(n \le 5000)$ 个不超过 10^6 的正整数组成的序列。请输出这个序列的**最长上升子序列**的长度。

最长上升子序列是指，从原序列中**按顺序**取出一些数字排在一起，这些数字是**逐渐增大**的。

输入格式

第一行，一个整数 n，表示序列长度。

第二行有 n 个整数，表示这个序列。

输出格式

一个整数表示答案。

输入输出样例　1 2 3 4

输入 #1 复制	输出 #1 复制
6 1 2 4 1 3 4	4

说明/提示

分别取出 1、2、3、4 即可。

B3637 最长上升子序列

$f(4)=3$

$1\ \boxed{f(j)+1} \Rightarrow f(i)$

设DP 数组 $f[i]$ = 以第 i 个数结尾的 LIS 长度

$f(j)$

转移:

j

$$f[i] = 1 + \max_{j<i\ and\ \boxed{a[j]<a[i]}}\{f[j]\}$$

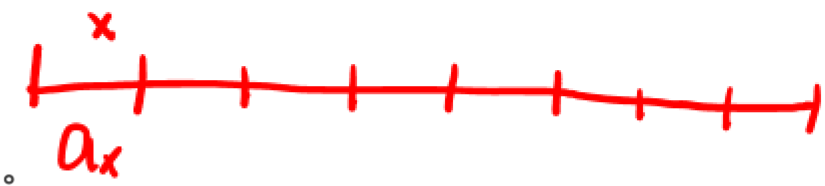
于是直接转移可以得到平方复杂度的算法。

P13015 [GESP202506 六级] 学习小组

j+1 ~ i

班主任计划将班级里的 n 名同学划分为若干个学习小组，每名同学都需要分入某一个学习小组中。观察发现，如果一个学习小组中恰好包含 k 名同学，则该学习小组的讨论积极度为 a_k。

给定讨论积极度 $a_1, a_2, \dots, a_n$，请你计算将这 n 名同学划分为学习小组的所有可能方案中，讨论积极度之和的最大值。



输入格式

第一行，一个正整数 n，表示班级人数。

第二行，n 个非负整数 $a_1, a_2, \dots, a_n$，表示不同人数学习小组的讨论积极度。

输出格式

输出共一行，一个整数，表示所有划分方案中，学习小组讨论积极度之和的最大值。

对于 40% 的测试点，保证 $1 \le n \le 10$。

对于所有测试点，保证 $1 \le n \le 1000$，$0 \le a_i \le 10^4$。

P13015 [GESP202506 六级] 学习小组

根据线性 dp 的想法， 考虑设 $f[i]$ 表示考虑前 i 个同学，讨论积极度之和最大值是多少。

$f(0)=0$

$$f[i] = \max\{f[i], f[j] + a[i-j]\}$$

$r-l+1$

$r-(l-1)$

$n \cdot n \quad O(n^2)$

枚举 j 转移即可。

$i-j$

$i-(j+1)+1$

4min + 10min 下课+思考　　3:00讲课

洛谷

P10721 [GESP202406 六级] 计算得分

abc a_1　abcabc a_2

abc

小杨想要计算由 m 个小写字母组成的字符串的得分。

abc abc

小杨设置了一个包含 n 个正整数的计分序列 $A=[a_1,a_2,\ldots,a_n]$，如果字符串的一个子串由 $k(1\le k\le n)$ 个 **abc** 首尾相接组成，那么能够得到分数 a_k，并且字符串包含的字符不能够重复计算得分，整个字符串的得分是计分子串的总和。

9

例如，假设，字符串 **dabcabcabcabzabc** 的所有可能计分方式如下：

2　1　0　1

- **d+abc+abcabc+abz+abc** 或者 **d+abcabc+abc+abz+abc**，其中 **d** 和 **abz** 不计算得分，总得分为 $a_1+a_2+a_1$。
- **d+abc+abc+abc+abz+abc**，总得分为 $a_1+a_1+a_1+a_1$。
- **d+abcabcabc+abz+abc**，总得分为 a_3+a_1。

小杨想知道对于给定的字符串，最大总得分是多少。

m^2x

mn

对于全部数据，保证有 $1\le n\le 20$，$1\le m\le 10^5$，$1\le a_i\le 1000$。

20×10^5

洛谷

P10721 [GESP202406 六级] 计算得分

可以发现，这同样是一个在序列上的问题，我们同样可以根据线性 dp 的想法，设 $f[i]$ 表示考虑了前 i 个位置可以获得的最大得分。

根据题意，分数和连续 abc 的个数有关，我们考虑枚举从 i 开始往前切分的这一段有多少个 abc。

$f(i)=\max\{f(i), f(j)+val(j+1, i)\}$

abcabc

k

$f(j)+a_k \to f(i)$

P10721 [GESP202406 六级] 计算得分

假如最后一段有 j 个 abc， 那么可以得到转移：

$$f[i] = \max\{f[i], f[i-3j] + a[j]\}$$

由于 $j \le n$，总复杂度是 $O(mn)$ 的。

$j > n$

背包问题

背包问题

背包问题是一类常见的 DP 问题， 其具备若干变式， 比如 01 背包，完全背包等。我们将在接下来的例题中具体介绍。

P1048 [NOIP 2005 普及组] 采药

辰辰是个天资聪颖的孩子，他的梦想是成为世界上最伟大的医师。为此，他想拜附近最有威望的医师为师。医师为了判断他的资质，给他出了一个难题。医师把他带到一个到处都是草药的山洞里对他说："孩子，这个山洞里有一些不同的草药，采每一株都需要一些时间，每一株也有它自身的价值。我会给你一段时间，在这段时间里，你可以采到一些草药。如果你是一个聪明的孩子，你应该可以让采到的草药的总价值最大。"

如果你是辰辰，你能完成这个任务吗?

输入格式

第一行有 2 个整数 T $(1 \le T \le 1000)$ 和 M $(1 \le M \le 100)$，用一个空格隔开，T 代表总共能够用来采药的时间，M 代表山洞里的草药的数目。

接下来的 M 行每行包括两个在 1 到 100 之间（包括 1 和 100）的整数，分别表示采摘某株草药的时间和这株草药的价值。

输出格式

输出在规定的时间内可以采到的草药的最大总价值。

P1048 [NOIP 2005 普及组] 采药

这是一个最典型的背包问题， 被称为 01 背包。

稍加转化描述题意：你有一个时间背包，容量是 T。面前三类选择：取某株草药/不取某株草药。最终希望把时间装满（或不超过）且价值尽量大。

①

②

③

P1048 [NOIP 2005 普及组] 采药

设 $dp[i][t]$：只考虑前 i 株草药、总时间为 t 能得到的最大价值。

转移：

不取第 i 株：$dp[i][t] = dp[i-1][t]$

取第 i 株（前提 $t >= w[i]$）：

$$dp[i][t] = \max(dp[i][t], dp[i-1][t-w[i]] + v[i])$$

答案：$dp[M][T]$。

时间/空间：$O(M*T)\,/\,O(M*T)$。

P1048 [NOIP 2005 普及组] 采药

重新回顾一下我们的问题，我们的问题可以简单描述为:

“有若干物品，每个物品要么选要么不选，每个物品有选择的代价和价值，我们希望在总代价不超过某个值的条件下最大化选择的价值”。

而我们的做法就是设计 dp 状态 dp[处理了前多少个][总代价]，然后枚举最后一个处理到的物品选还是不选来转移。

这就是 01 背包的普遍套路。

为什么叫 01 背包呢？这是因为选/不选对应 1/0 。

P1048 [NOIP 2005 普及组] 采药

写 dp 的时候， 有一个叫滚动数组的技巧。因为 dp 数组空间占用其实很大， 所以实际应用的时候， 一般会把第一个维度滚动掉来优化空间。

对于 01 背包， 我们的处理方法就是直接删去第一维， 然后转移的时候注意把第二个维度倒序循环。

P1049 [NOIP 2001 普及组] 装箱问题

有一个箱子容量为 V，同时有 n 个物品，每个物品有一个体积。

现在从 n 个物品中，任取若干个装入箱内（也可以不取），使箱子的剩余空间最小。输出这个最小值。

输入格式

第一行共一个整数 V，表示箱子容量。

第二行共一个整数 n，表示物品总数。

接下来 n 行，每行有一个正整数，表示第 i 个物品的体积。

输出格式

- 共一行一个整数，表示箱子最小剩余空间。

输入输出样例

输入 #1 复制

```
24
6
8
3
12
7
9
7
```

输出 #1 复制

```
0
```

说明/提示

对于 100% 数据，满足 $0 < n \le 30$，$1 \le V \le 20000$。

P1049 [NOIP 2001 普及组] 装箱问题

总体问题和上一题差不多，问题在于要求的最大价值变成最小化剩余空间。

bool

更改 dp 值的定义，让 $dp[i][j]$ 表示考虑前 i 个物品，能否选出若干个占用箱子空间为 j 即可。

物品

x

$$dp[i][j] \mid= dp[i-1][j-x]$$

W_1+W_2 A

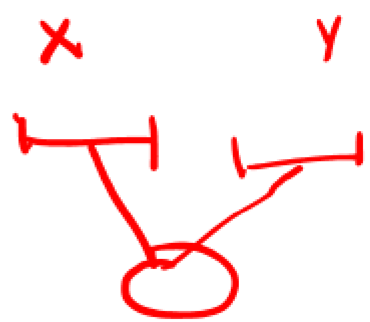
W_3 W_4 B

$W_1+W_2-W_3-W_4$

P8742 [蓝桥杯 2021 省 AB] 砝码称重

你有一架天平和 N 个砝码，这 N 个砝码重量依次是 $W_1, W_2, \cdots, W_N$。请你计算一共可以称出多少种不同的重量?

注意砝码可以放在天平两边。 砝 左/右



输入格式

输入的第一行包含一个整数 N。

第二行包含 N 个整数: $W_1, W_2, W_3, \cdots, W_N$。

$|x-y|$

输出格式

输出一个整数代表答案。

$f(i,j)$ $f(n,x)$

对于 50% 的评测用例，$1 \le N \le 15$。

对于所有评测用例，$1 \le N \le 100$，N 个砝码总重不超过 10^5。

P8742 [蓝桥杯 2021 省 AB] 砝码称重

其实还是 01 背包问题， 只是砝码可以选择其重量是正数还是负数。

在转移的时候两者都考虑一下即可。

$$f(i,j) \mid= f(i-1,j)$$

$$\cancel{= \max\{ f(i-1, j-w_i}$$

$$f(i,j) \mid= f(i-1,j-w_i)$$

$$f(i,j) \mid= f(i-1,j+w_i)$$

4.0- 上课

P1616 疯狂的采药

有 m 种草药， 一共可以用来采药的时间是 t。采摘一份第 i 种草药需要花费时间 a_i, 同时可以获得 价值 b_i。每种草药都有无数份。

请问你最多可以采价值为多少的草药？

$$1 \le m \le 10^4, 1 \le t \le 10^7, m \times t \le 10^7, 1 \le a_i, b_i \le 10^4$$

P1616 疯狂的采药

考虑认为时间就是背包容量， 可以发现这和 01 背包问题很相似，唯一的区别是， 这里每个物品有无数个。而在 01 背包中， 每个物品只有一个。

完全背包

$$f_{i,j}=\max\{f_{i-1,j},\ \boxed{f_{i-1,j-w_i}+v_i,\ f_{i-1,j-2w_i}+2v_i,\ \dots}\}$$

$$f_{i,j-w_i}=\max\{f_{i-1,j-w_i},\ f_{i-1,j-2w_i}+v_i,\ \dots\}.$$

我们还是设 $f[i][j]$ 表示只能选前 i 个物品的时候可以得到的最大值。

一个简单的想法如下：$\lfloor \frac{j}{w_i} \rfloor$

n^3

$$f_{i,j}=\max_{k=0}^{+\infty}(f_{i-1,j-k\times w_i}+v_i\times k)$$

但是这样复杂度很高。 n^3

完全背包

f

但是其实有简单的方法：　　　j　$j-w_i$

$\cancel{f_{i-1,j-w_i}}$

$$f_{i,j} = \max(f_{i-1,j}, \boxed{f_{i,j-w_i}} + v_i)$$

理由是当我们这样转移时，$f_{i,j-w_i}$ 已经由 $f_{i,j-2\times w_i}$ 更新过，那么 $f_{i,j-w_i}$ 就是充分考虑了第 i 件物品所选次数后得到的最优结果。换言之，我们通过局部最优子结构的性质重复使用了之前的枚举过程，优化了枚举的复杂度。

这样的话复杂度和 01 背包一致了。

完全背包

完全背包也可以滚动数组优化。和 01 背包唯一的不同在于我们在转移的时候是顺序循环的。

P11246 [GESP202409 六级] 小杨和整数拆分

i^2

小杨有一个正整数 n，小杨想将它拆分成若干完全平方数的和，同时小杨希望拆分的数量越少越好。

编程计算总和为 n 的完全平方数的最小数量。

$n, \quad i, \quad i^2, \qquad i^2 \le n$

$i \le \sqrt{n}$

$j: 0 \sim n$

$f(j) = \min\{f(j),\ f(j - i*i) + 1\}$

输入格式

输入只有一行一个正整数 n。

输出格式

输出一行一个整数表示答案。

输入输出样例

输入 #1	输出 #1
18	2

$\sqrt{n} \cdot n \qquad O(n\sqrt{n})$

说明/提示

数据规模与约定

对全部的测试数据，保证 $1 \le n \le 10^5$。

P11246 [GESP202409 六级] 小杨和整数拆分

背包容量为 n，第 i 个物品重量 i^2，每个物品有无数个，标准的完全背包问题。

需要枚举 $\sqrt{n}$ 个数，每个数 dp 的复杂度是 n，总复杂度 $O(n\sqrt{n})$

总结

f(i)

f(i)

洛谷

总结

我们今天主要复习了简单的线性 dp、01 背包和完全背包。

这三者是动态规划中最基础的部分， 关键点在于抓到问题的关键特点、关键条件， 然后"向标准模型进行转化，"从而得出解法。

这里的背包问题尤为重要， 希望大家可以牢记背包问题的解法与特点。

Thanks!