

动态规划

wwwodddd

1 自我介绍

毕克

2010 ~ 2013, 石家庄二中。

NOI 2012 金牌。

2013 ~ 2017, 清华大学。

5 次区域赛/EC-FINAL 金牌。

没去过 IOI, 也没去过 ACM World Final。

三大爱好: 比赛, 出题, 讲课。

My vegetable exploded up.

QQ:751723392, Email:wwwodddd@gmail.com

2 二分法

主要分为整数的二分, 实数的二分。

2.1 整数二分

整数二分主要是结束条件的设置, 这还关系到取整部分。

一个比较不容易出错的方法是使用 $L < R - 1$ 。

这样无论中间二分的除以 2 是如何取整的, 都不会导致死循环。

2.2 实数二分

根据实现经验, 实数二分一般控制二分次数较好。

每二分 10 次, 可以获得大约 3 位精度 ($1/1024$)

`double` 在二进制下, 小数点后有 52 位。

3 wqs 二分

王钦石浅析一类二分方法

求选择 k 个的最优解, 不方便直接求解。

二分一个权值 v , 使得每选一个, 可以额外获得 v 的权值。

显然 v 最优解中的个数是单调的。

二分 v 来确定最优解。

4 bzoj 2654

4.1 题目背景

bzoj 2654 tree

4.2 题目大意

4.3 题目分析

5 基础知识

动态规划概念

1. 可以转化为子问题。
2. 最优子结构，子结构最优，必然全局最优。如果不满足最优子结构，那么需要记录更多信息。
3. 无后效性，之前的操作不影响之后的过程。如果有后效性，那么需要记录更多信息。
4. 子问题有重复。记录下来，只做一次。如果没有重复，那么是否动态规划无所谓。

简单来说

1. 动态规划记录的东西必须足够，当然记录的越多，程序越慢；最后就成了暴力。
2. 不能有环，动归顺序必须是有向无环图，少部分题目会有自环。

两个基本问题

1. 最优化问题。
2. 计数问题。

两种状态表示方法

1. 起点到当前的情况。多数情况常用。
2. 当前到终点的情况。算期望题目常用。

实现相关

1. 枚举谁转移到自己。简单题，记忆化搜索。
2. 枚举自己转移到谁。复杂题，状态压缩。

6 poj 1163

6.1 题目背景

poj 1163 The Triangle

6.2 题目大意

6.3 题目解法

7 最长不下降子序列和最长公共子序列

7.1 最长不下降子序列

输入一个长度为 n 的序列，求最长不下降子序列。

解法的话，常见的有两种。

第一种 $O(n^2)$ 暴力 DP， f_i 表示以第 i 个元素结尾，最长不下降子序列是多长。可以略加优化，设 f_i 表示以权值 i 结尾的最长不下降子序列是多长。这样修改和询问操作都可以用线段树维护，也可以得到一个 $O(n \log n)$ 的做法。

第二个 $O(n \log n)$ 贪心， f_i 表示长度为 i 的不下降子序列，结尾最小是多少。

7.2 一些变形

最简单的变形就是改成最长上升子序列，基本等于没改。

然后就是问，最少由几个不下降子序列覆盖。

答案是最长下降子序列的长度（翻转，最长上升子序列的长度）

[Luogu P1020](#)

8 最长公共子序列

注意子序列和子串的差别。

输入两个字符串，问最长公共子序列的长度。

解法的话，常见的有一种，用 $f_{i,j}$ 表示第一个串的前 i 个字符，和第二个串的前 j 个字符的最长公共子串的长度。时间复杂度 $O(n^2)$ 。

8.1 一些变形

输入两个排列，求最长公共子序列。

经过替换后，等价于求最长不下降子序列，时间复杂度从 $O(n^2)$ 变为 $O(n \log n)$ 。

输入两个序列，求最长公共子序列，保证每个数字在每个序列中至多出现 $O(1)$ 次。

将第二个字符串中每个数字，换为第一个字符串中出现的所有位置的倒序，求最长不下降子序列。

8.2 另一些推广

三维最长不下降子序列（之前提到的是二维）[BZOJ 2253](#)

大概两种做法，第一种是 CDQ 分治，第二种是数据结构维护，其中 CDQ 分治较为好写。

四维最长不下降子序列。

CDQ 分治两次。

8.3 百度之星 2018

在我反复讲这个题目和变形之后，百度之星 2018 出了 [hdu 6412](#) 送我进了百度之星决赛
百度之星决赛办的非常荒诞[如何评价](#)

9 hdu 6412

9.1 题目背景

[公共子序列](#)

9.2 题目大意

9.3 题目解法

转化为 k 维严格下降序列的个数。
转化之后的序列长度，期望为 n 。
直接暴力即可。

10 51nod 1376

10.1 题目背景

[51nod 1376 最长递增子序列的数量](#)

10.2 题目大意

10.3 题目分析

11 Luogu P3958

11.1 题目背景

[P3958 奶酪](#)

11.2 题目大意

11.3 题目解法

许多人懂得通过两边平方避免开根号。
但是这个题数据范围比较大，两边平方后需要通过 `unsigned long long` 避免越界。

```
1
2 10000000000 1
1000000000 1000000000 1000000000
-10000000000 -10000000000 -10000000000
```

这份题解

因为两边平方，避免开根号，距离越界成负数就 GG 了
但是另一方面，开根号会有误差

```
1
2 1000000000 500000000
0 0 0
1172 44706 999999999
```

这个数据可以搞掉转 double 开根号的

题解

因为三维的平方和是 10000000000000000021

开根号就是 1e9

同样卡掉了这份题解

题解

题解

12 Luogu P3959

12.1 题目背景

P3959 宝藏

距离 NOIP 还有 2 个月，相信大家都做过。

12.2 题目大意

12.3 题目解法

这个题目的数据非常弱，以至于各种错误的贪心，直接搜索都可以通过。

大家可以试图卡一下这份题解中的做法。

在评论区有热心观众给出了数据。

13 背包 DP

主要分为三种

- 零一背包
- 完全背包
- 有限背包

14 树形 DP

树形 DP 一般较为简单，因为可以轻易的估计出每个点状态的个数。

15 bzoj 2152

15.1 题目背景

CTSC2006 聪聪可可

15.2 题目大意

输入一个树，有边权。

问有多少条路径长度是 3 的倍数。

15.3 题目分析

很多人写点分治，完全不知道为什么。

让每条路径在最浅的点被统计即可。

16 Codechef BLACKCOM

16.1 题目背景

Black Nodes in Subgraphs

16.2 题目大意

输入一个树，问是否存在一个子联通区域，大小为 S ，有 B 个黑点。

多组询问，点数 $N \leq 5000$ 。

16.3 题目解法

预处理，回答所有询问。

对于每个 S 求出最大可能的 B 和最小可能的 B ，则中间的所有 B 都可以达到。

对于最大最小来说就是一个背包问题。

虽然看起来很奇怪，但是这个背包时间复杂度是 $O(N^2)$ 的。

17 有向无环图 DP

18 bzoj 3036

18.1 题目背景

bzoj 3036

18.2 题目大意

一个有向无环图，从 1 走到 n ，每次随机选一条出边走。

问期望路径长度是多少。

18.3 参考题解

别人的题解

事实上直接 DP 就好了。

注意到 n 是唯一一个出度为 0 的点。

19 数位 DP

数位 DP 作为计数类 DP，对拍一般比较容易。

20 ProjectEuler 156

20.1 题目背景

Counting Digits

20.2 题目大意

设 $f(n, d)$ 为 1 到 n 中， d 出现了多少次。

对于每个 d 求出所有满足 $f(n, d) = n$ 的 n 。

20.3 题目分析

首先实现出 $f(n, d)$ 函数。

设 $t = |f(n, d) - n|$ 那么下一次 $f(n, d) = n$ 出现的位置，至少要在 $t/11$ 之后。

如此暴力即可。

21 状态压缩

简单的状态压缩，可以优化许多动态规划。

22 ProjectEuler 222

22.1 题目背景

Sphere Packing

22.2 题目大意

22.3 题目分析

需要找到一个叠放的顺序。

设 $d(i, j)$ 表示第 i 球和第 j 个球错开放，比叠起来放，距离节约多少。

相当于是求哈密顿路，距离最长。

50, 48, 46, 44, 42, 40, 38, 36, 34, 32, 30, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49 是最优解。

23 hdu 6413

23.1 题目背景

棋盘上的旅行

23.2 题目大意

23.3 题目解法

随机染色，状态压缩 DP。

每次成功的概率是 $\frac{k!}{k^k}$ 。

随机几百次，就有概率通过了。

24 bzoj 1494

24.1 题目背景

[NOI2007] 生成树计数

24.2 题目大意

n 个点构成一条链，如果两个点之间的距离 $\leq k$ ，那么他们之间有边。

问生成树个数。

24.3 题目解法一

当时的官方解法大概如下。

如果会行列式对质数取模，用高斯消元求值，便可以得到 60 分的高分。

第 7 个点和第 8 个点本意是面相写出 $O(n)$ 状态压缩的选手的，但是实际上行列式写得好也能过。

如果想通过这个题，需要想到状态压缩的做法。

考虑存下最后 k 个点的连通性，对于 $k = 2, 3, 4, 5$ 分别有 2, 5, 15, 52 种可能。

这个是 **贝尔数**，或者说是第二类斯特灵数的求和。

之后转移就枚举下一个点和之前 k 个点的连接情况共 2^k 种可能性。

不能出现环。

如果即将抛弃的点和其他 $k - 1$ 个点均不连通，那么必须和新加进来的点连通。

生成转移矩阵，转移即可。

24.4 题目解法二

出题人以为题目中抛出一个看似无用的行列式，可以转移大家焦点，隐藏题目真实解法。但是实际上这个题在近年有了新的做法，行列式的计算复杂度是 $O(n^3)$ 。

通过行列式可以得到前几百项的结果。

设想这个答案可以通过矩阵乘法求出，那么他必然是一个 ≤ 52 阶的递推数列。

不妨设递推数列为

$$f_n = \sum_{1 \leq i \leq d} a_i f_{n-i}$$

你可以直接用前 $2n + 1$ 项，列出 n 个方程，解出答案，但是略微麻烦。
解方程，除了解出唯一解，还有两个可能，无穷多解和无解。