2010/4/18

ADVENTOPLAB

USACO 第四章通关总结

全面的总结 | ADVENTop

USACO 第四章通关总结

By ADVENTop

第四章相比较于前三章是一个不错的进阶，前三章重在基础，第四章则加入了更多精妙的算法思想，和更多细节的推敲。整体感觉提高很大，这章开始对于初学者来说，积累了相当的经验，引入了很多算法和对于曾经的经典模型的再度理解，可以说对于自身的技术和算法有了长足的进步。本章学到的算法有：部分**定理，网络流，二分图匹配，拓扑排序深度实现，最小割方案**，同时巩固提高了**剪枝**技术，浅尝了高等的**构造**。

第四章题目总体类型分布:

Section 4.1	DONE	2010.02.09	PROB Beef McNuggets 斐蜀定理+重复背包
	DONE	2010.02.17	PROB Fence Rails 背包的搜索（强力剪枝）
	DONE	2010.02.19	PROB Fence Loops 最小环+构图
	DONE	2010.02.23	PROB Cryptcowgraphy 搜索+强力剪枝
Section 4.2	DONE	2010.03.10	PROB Drainage Ditches 网络最大流
	DONE	2010.03.15	PROB The Perfect Stall 二分图最大匹配
	DONE	2010.03.28	PROB Job Processing Johnson 划分
	DONE	2010.03.30	PROB Cowcycles DFS+数学统计
Section 4.3	DONE	2010.04.04	PROB Buy Low, Buy Lower LIS+判重
	DONE	2010.04.06	PROB The Primes 构造+枚举
	DONE	2010.04.07	PROB Street Race DFS（时间戳思想）
	DONE	2010.04.08	PROB Letter Game 位运算+检索
Section 4.4	DONE	2010.04.10	PROB Shuttle Puzzle 构造
	DONE	2010.04.10	PROB Pollutant Control 最小割方案
	DONE	2010.04.11	PROB Frame Up 拓扑排序

◎第一节:

这一节的题目更像是对于前三章的总结，前三章做的最频繁的就是搜索，但是对于剪枝来说，大部分都模棱两可，有的甚至可以枚举出解。但是到了第 4 章情况就大不相同了。剪枝在这一节达到了最高峰，很像模像样，一如既往的经典，包括精妙的思维模式，对于任何水平的人来说，都有你能得到的东西。

4.1.1：本题目的大意是找到不可能组合成的数字，可以使用重复背包，但是本题目的难点却不在背包本身，而是在于如何限定上界。这就要用到斐蜀定理：1.所有数字的最大公约数不是 1，则必定会有一个很大的数字不可被整除，无解。2.两个素数的乘积之后的数字均可以被这两个数字以任意整数倍得到。有了这些，就很容易判定无解和限定上界了。

4.1.2：经典的背包模型，而且是多个背包，很类似上一章的最后一道题目，可是本题目顺序

无关，所以需要搜索求解。题目的规模不小：有 50，直接搜索固然不行。所以本题目可以用 DFS-ID 试探着进行，也就是先使数据有序化，这样可以方便剪枝。然后一层一层迭代，当第 I 层不可行时，第 I+1 之后必然不可行，所以就省去了大量时间。搜索时用板材作为深度，原料作为背包进行搜索。然后要考虑剪枝：1.可行性剪枝：考虑引入一个概念，浪费指数。如果这个指数比最小板材的好要小，那么后面的必然可以不用搜索了。2.最优性剪枝：因为题目板材有 1023，但是板材最长不过 128，所以必定会有一样长的板材，数据是稠密的，搜索时假设搜到了第 J 个背包，那么我们对于一样长的板材不必要在从头尝试（已经有序化了），而是接着第 J 个背包继续搜索。这两个剪枝可以使时间大大减少。最后因为我们提前排了序，所以本题目可以转化为判定性问题，使用二分答案求解。

4.1.3：本题目难在构图，构图没什么规律，朴素即可。然后 DFS 探索无重复路径（必须至少找到 3 个点），更新最小环。

4.1.4：本题目很考察思路的条理性，首先我们有一个目标状态，然后有一个初始状态，没有确定的规律，所以要搜索。难在如何设计剪枝。这里有 3 个角度：1.重复状态的避免，可以使用 Hash 来快速完成这个使命.2.搜索顺序的选择，因为中间的字母”o”决定着整体的平衡，所以要先枚举”o”的位置，然后是”c”,”w”。3.可行性剪枝：通过观察,”cow”中间的子串无论怎么变都不会改变顺序，所以对于”cow”中间的模式串，匹配一下看看他是否在目标串中。不在则剪枝。有了这些优化，搜索就很快了。

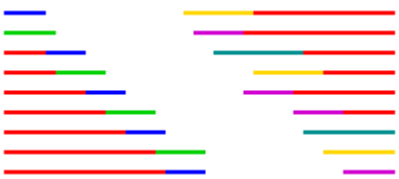
◎第二节:

这一节引入了网络模型，开始进入高级图论的分析阶段，很考验建模能力，特别是对于细节的分析，有些题目的思考模式也发人深思，不同于正常的套路，换个角度就有意想不到的结果。

4.2.1：本题目的描述中暗含着一个标准的网络流模型，直接建模求出流量即可。但是题目还提示可能出现环，这是针对于 E-K 而言的，因为 E-K 法会陷入死循环。推存使用 Dinic,ISAP 这样的性价比高的算法，或者是 HLPP.

4.2.2：由于题目说明了一个栅栏只可以存在一头牛，所以这是一个匹配问题，因为只有牛与栅栏两个对象，所以可以建模成二分图，然后使用匈牙利算法出解。

4.2.3：一道经典的贪心问题：首先问题可以无关 A，B 的先后，对于每一个任务贪心的求出最小值，然后用 A 中最小的与 B 中最大的相加，选出最小的，这就是 Johnson 划分，证明略。



思想非常精妙，因为最大的空闲时间中存在的任务耗时越多，相对的也就越省时间。

4.2.4：本题目考察数学统计，直接 DFS。题目给出的齿轮间的关系就是可行性剪枝的条件。

◎第三节:

这一节题目类型很自由，更像是考察了一些自由的类型的题目，这是比赛中比较常见的类型，更接近实战，而且可以使选手积累大量实战经验。融合了很多常见的优化和编程细节。

4.3.1：经典的 LIS,但是还要输出方案总数，也就是需要统计，并且需要判重。实现时使用一个线性表存储当前最优序列。使得求 LIS 优化为 O(Nlog2N)。接着要求不重复方案。思路很简单：就是对于同一个阶段的数字进度统计一次。因而就有两个方法：1.用一个 bool 数组记录每个阶段出现过的数字，用以判重。2.对于一个数字，我们记录他的**最近**的位置，检索时不断更新，计算方案时只统计最近的，我把这个命名为**单调保留模式**。这个思想很有用途，对于判重很实用。

扩展：2008 天津市省级选拔赛试题。

【题目描述】

一个字符串的子串是在这个串基础上去掉 0 个或者若干个字符所形成的，例如 abc, aa 和 abbc 都是字符串 aabbcc 的子串，而 aba 不是。 现给你三个字符串，请问他们三个共同含有多少种子串（不算空串）？

注意： 有些相同的公共子串尽管出现在不同的位置，但仍算 1 种，详见样例。

【输入】

每组测试数据只有 3 行，每行都是一个只包含有小写字母的字符串。

【输出】

输出 3 个字符串共有的公共子串种类数。

【样例输入输出】

sub.in

apartment

apache

approach

sub.out

6 {“a”, “p”, “ap”, “pa”, “aa”, “apa”}

分析：就是最朴素的 DP 进行统计，而且要分阶段，那么如何判重呢？对于一个序列的相同字母，我们倒序扫描，扫到 I 位置，我们只统计最近的一个字母，也就是单调保留，这样问题就迎刃而解了。因为题目数值很大，需要高精度，可以借助本题目练习高精度的运算符重载。

4.3.2：因为题目的限定很多，所以可以使用优化后的枚举，当然可以搜索。我的方法算是一个创新，构造出所有可行的方案，然后分阶段枚举：1.对角线。2.四周。因为已经构造好了，可以二分查找。需要一些剪枝：1.可行性。2.奇偶性。3.0 的去除。这样就可以在规定的时间出解了。这个方法的优点是：代码短，结构清晰，好调试。但是却牺牲了一定的时间。对于实战来说，利大于弊。

4.3.3：本题目要求出一个特殊的分界点，可以枚举删除的点，然后 DFS 判断。第二问是第一问的子集，先构造出分层图，然后搜索，利用时间戳的概念，只要找到点的层次浅于自己，则不合法。直到找到所有。

4.3.4：本题目是一道黑箱字典题目，因为有序了，所以二分答案成为可能。但是使用为运算优化一下，也可以很快速出解，机理是：对于给定的可用字母（最长 7 位），用一个 7 字节以上的数字表示使用的字母，这样可以在 O(1)的时间内判断是否可以连接。首先先要枚举所有单词，找到最大的长度。然后用单调的顺序扫描，判断可行性的方式是按位与，如果为零则可行。但是他们的位置数字可能冲突，其实只要从两个方向分别记录一个就行了，这利用了贪心的性质。

◎第四节:

这本节的题目比较困难，像是提前过渡到了第 5 章，首次出现了专门的构造类题目，图论的分析再次加深，对于题目的理解难度也加大，这不失为一种不错的进阶，到了这里可以说，对于联赛难度的问题已经终结，而对于竞赛则开始入门了。

4.4.1：本题目是很经典的黑白棋子移动的变种，朴素的方法是：观察规律，题目有对称性，所以可以 BFS 到一半，后面的构造出来。但是本题目本身的答案就是一个对称数列，分析观察可以得知：答案为长度是 1,2,3,4,...,n,n+1,n,...,4,3,2,1 这样的 2n+1 组等差序列，奇数的为递减，偶数的为递增。公差是二，偶项的首相为 n-i+1,奇项是 n+i.后面的可以继续根据对称性构造。

4.4.2：比较明显的最小割方案：先求出最大流，根据最大流最小割定理，每次枚举饱和边，并将其删除，再次查看网络流是否恰好变化了被删边的流量，是则继续，否则回溯。这样可以求出一组最小割方案。查找时，分为两类：一类是最小割是一条割边，这样直接按照顺序遍历即可。另一类是最小割多于一条，同样按顺序遍历，保证题目要求。

4.4.3：本章的压轴题目，本身题目不难，但是题意理解和建模较难。首先：题目明确说明每个矩形可以看见，这样很容易得知一点：在一个矩形上，若有不是本身的字母出现，那么非自身的字母所代表的矩形一定是压在当前矩形上面的，这样可以建立拓扑关系。因为题目要求输出所有解，所以需要使用 DFS 统计，最后需要排序。

★章末总结：

这一章，我个人感觉收获颇丰。无论是前三章搜索的总结，强大的剪枝。还是网络模型的引入，精妙的贪心思想，直到一些优化技巧。最主要的经验就是**换个角度**看待问题。图论问题是对问题模型的一种**看法**，这种看法有时不止一种，看法不同，问题的解法也就不一，但是往往问题的解法比较集中，这样，一个好的“看法”就通常起到**决定算法好坏**的**关键**地步。所谓看法，其实是对问题模型的理解，你的理解深入，就能使用好的数据结构去描述题目。相应的算法也能抉择出更加高效的。所以，锻炼自己看法的好坏是十分重要的，这就需要扎扎实实的练就基础，以能做到深入剖析问题的能力，对一个问题的深入剖析，是我这章的最大收获。

2010/4/18