

2010

USACO 第三章通关总结

全面的总结

[HTTP://hi.baidu.com/adventop](http://hi.baidu.com/adventop)



USACO 第三章通关总结

By ADVENTop

第三章总体上感觉并不太困难,但是有很锻炼思想的题目,而前两章只要想一想就能拥有好的解答,但是第三章就不行了,你必须深思熟虑考虑最优的,也正是如此,它比第二章有了一个提高.具体的说第三章的训练循序渐进,有条不紊,无论是题目质量还是考察要点,均很不错.同时也新加入了一些相对高等的算法.对于本章我学到的主要算法有:Prim, 多种背包 DP, 计算几何初步, 欧拉路径, 高斯消去法, 皮克定理. 并且很好的巩固了我的技术和基础.

第三章题目总体类型分布:

Section 3.1	DONE	2009.10.12	PROB Agri-Net Prim 求 MST
	DONE	2009.11.07	PROB Score Inflation 重复背包
	DONE	2009.12.05	PROB Humble Numbers 枚举
	DONE	2009.12.26	PROB Shaping Regions 矩形分割 (分治)
	DONE	2009.12.31	PROB Contact 位运算优化枚举
	DONE	2010.01.22	PROB Stamps 特殊背包
Section 3.2	DONE	2010.01.22	PROB Factorials 优化计算
	DONE	2010.01.23	PROB Stringsobits DP+枚举
	DONE	2010.01.24	PROB Spinning Wheels 染色+枚举
	DONE	2010.01.29	PROB Feed Ratios 高斯消元
	DONE	2010.01.30	PROB Magic Squares BFS+Hash
	DONE	2010.01.31	PROB Sweet Butter SPFA
Section 3.3	DONE	2010.01.31	PROB Riding The Fences 欧拉路径
	DONE	2010.02.01	PROB Shopping Offers 泛化多维背包
	DONE	2010.02.02	PROB Camelot BFS+枚举
	DONE	2010.02.02	PROB Home on the Range DP or 枚举
	DONE	2010.02.03	PROB A Game DP
Section 3.4	DONE	2010.02.05	PROB Closed Fences 计算几何(2 分法)
	DONE	2010.02.06	PROB American Heritage 树结构(分治)
	DONE	2010.02.06	PROB Electric Fence 皮克定理
	DONE	2010.02.06	PROB Raucous Rockers 双重 DP or 多维 DP

◎第一节:

这一节的题目主要是对于第二章的衔接,因此他的题目难度都不是太大,或者说没有涉及高深的算法,但是要求的技术思想却进步了,特别是分析问题的能力要有一个提高,也正

是这样,这节有很优秀的题目.

3.1.1: Agri-Net最短网络:这道题目就是初步的MST,MST对于NOIP是很重要的算法,MST的出现也意味着图论的分析要求要加大,不过本题并不困难.

3.1.2: Score Inflation 总分:这是一个重复背包,给定一个有限容量的背包,在有限容量范围内,选取最大的价值组合.因为每个物品可以重复选取,所以有 $DP[i][j]=MAX\{DP[i-1][j], DP[i][j-V[i]]+W[i]\}$,请大家注意方程第二个分部.而本题目的模型恰可以转化到这上面来,每个“物品”可以多选,所以是重复背包.

3.1.3: Humble Numbers 丑数:首先这是一道枚举,不过是需要优化的枚举,分析题目我们知道,构造丑数是必要的,因此我们需要在构造上优化.假定我们已经构造出 $\{a_1, a_2, a_3, a_4, \dots\}$ 的一个序列,他是严格有序的那么,这个我们知需要找一个足够小的基丑数乘上这些序列的数,找到比 a_n 大的最小的那个,这必定是 $a_{(n+1)}$ 个丑数,因此只要找到一个我们就可以记录以下当前位置,让其记住位置是因为假设我们已经计算到 $x \cdot a_4$ 才能满足,那么下次就没必要再从 a_1 到 a_3 了,因此可以立即剪枝大大优化时间效率.

3.1.4: Shaping Regions 形成的区域:这是一道好题,曾令无数初学者纠结,首先我们分析,后面的一定覆盖前面的.因此,我们只要倒序染色,自然就不会再去考虑覆盖的问题,接着我们没有必要把每个点都计算出来,因此可以离散化,用大的矩形表示,分析题目染过的部分(倒序)别的矩形就不可能再染,因此我们也不需要全盘枚举,只需考虑没有用到的部分,这样就可以切割矩形,把矩形切割成无数小的块,每次只要和这些没有用过的块比较就行了.这样递归下去,直到处理完所有.

3.1.5: Contact 联系:分析问题的规模,不大,可以直接枚举,因为要表示状态,可以使用位运算优化,然后 KISS 原则枚举就行了.

3.1.6: Stamps 邮票:这是一道经典的好题,乍一看就是一个多重背包,但是他的价值上限很大,因此单纯的背包找连续值,必定要超时.这是因为我们把重点放在了所有的情况可行不可行上,事实上我们把目标转化一下,找到第一个不满足的,也就意味着找到了解,这样我们不需要都算出来,也不要过多地调整(如果是背包,需要很多调整),这样时间上就很高效了, $DP[j]=MIN\{DP[j-V[i]]\}+1$,意思是当凑成 j 时需要的最少邮票数,当第一个 j' 超过限定时,输出 $j'-1$ 即可.

◎第二节:

这一节的题目就满足第三章要求了,仍然是巩固基础算法,但是题目的难度和需要的技术含量明显加大,比如单纯的算法无法解释,要有组合的算法,BFS+Hash , Dijkstra+Heap 等等,这样使得题目更具挑战.

3.2.1: Factorials阶乘:一上来,一道需要时空优化的经典题目,首先, $4220!$ 的计算空间上是无法承受的,而且时间也无法保证,但是分析题目,我们只需要保留最后的有效位因此可以极大地简化运算,再分析,后面的0是2,5两个因子构成的,因此把成对的2,5去掉就行了,这样的好处是计算大大简化,同时不依赖过多的空间.

3.2.2: Stringsobits01串 Kim Schrijvers:这是一道优化题目,直接枚举时必定不可行的,首先我们计算出 $dp[i][j]$ 表示*i*个长度的01串构成'1'不超过*j*的有多少个,可以用背包计数解决,然后分析得知如果要求得的那个数字的排名在 $dp[i][j]$ 之后,那么(*i*+1)位的数字必定是1,因为他在之后,反之就是0.这样枚举的复杂度是 $len(\text{数字长})$,可以说非常小了.

3.2.3: Spinning Wheels 纺车的轮子:分析周期,转过 360 次之后又回到原位,因此,我们只需枚举 0 到 359,这样的规模很小了,可以染色判断.

3.2.4: Feed Ratios 饲料调配:简易的高斯消元,其实是 3 元 1 次方程组,主要是解方程,控制精度 .

3.2.5: Magic Squares 魔板:可以双向 BFS,但是对于序列的顺序处理要合理,而本题目的给定序列是基于单向的,这时就要使用 Hash 优化了.可以应用 Cantor 展开,但是我是直接 Mod 一个奇数的简单的 Hash.

3.2.6: Sweet Butter 香甜的黄油:SPFA 或者 Dijkstra+Heap,因为规模很大,所以 Floyd 是不行的,这时候就要使用前两者了.

◎第三节:

这节更进一步,优化的细节更大,问题也更加困难,同时也加进了新的算法——欧拉路径 , 欧拉路径使得对于图论问题的分析更加深度,其实对于图论,建模的程度会使问题得难度下降,好的模型,也往往取决于你对于问题的理解.

3.3.1: Riding the Fences骑马修栅栏: 对于一个无向图, 如果它每个点的度都是偶数, 那么它存在一条欧拉回路; 如果有且仅有2个点的度为奇数, 那么它存在一条欧拉路; 如果超过2个点的度为奇数, 那么它就不存在欧拉路了. 这是基本的欧拉路径问题,欧拉路径的建模有时要把点与边进行相应的转化, 不然很有可能会变成Hamilton回路,那就很困难了.

3.3.2: Shopping Offers商店购物:一道泛化多维背包问题,是5维的,之所以泛化是因为,此类背包的物品往往是一组而不是单独的一个,不过实现起来主要是预处理难度加深,背包本身不难.

3.3.3: Camelot 亚瑟王的宫殿:经典问题,首先分析题目,我们不知道要在哪里汇合,骑士该如何与国王走,包括哪个骑士跟国王走,这些因素集合在一起,使得问题的不定性很大,也就是问题的规模很大,因此都需要枚举,但是为了降低时间复杂度,我们需要进行构造:先构造所有骑士的最短路,接着构造国王的,然后是国王与哪个骑士在哪个格子会合距离短,同时还要算出不包含国王的所有骑士到达一个格子的最短路径.所有的事情都完成后比直接枚举降低了好几个数量级,可以说很快了,这样我们是以牺牲空间换取时间.最后按照公式 $MIN(dis[x][y]+kdis[len'].val+当前骑士从当前会合点到(x,y)的距离-当前骑士到(x,y)本身的距离)$ 枚举一遍输出最小值即可,也就是构造所有需要用的,保证不重复计算.

3.3.4: Home on the Range 家的范围:必须要认真审题,才能看出枚举的精妙,首先直接枚举超

时是肯定的,那么我们怎么才能优化呢?观察样例,如果一个图形是矩形,那么他的所有格子必定都是 1,这样我们可以记录 $DP[i][j]$ 表示从(1,1)到(i,j)的所有 1 的个数,然后满足方程如下的 $sum[i+len-1][j+len-1]+sum[i-1][j-1]-sum[i+len-1][j-1]-sum[i-1][j+len-1]=len^2$ 就一定是符合的图形了,当然我们还可以更优,使用 DP:

```
for(int i=2;i<=n;i++)
    for(int j=2;j<=n;j++)
        if(map[i][j]=='1'){
            dp[i][j]=MIN(dp[i-1][j],dp[i][j-1],dp[i-1][j-1])+1;
            if(dp[i][j]>=2) square[dp[i][j]]++; //一边 DP 一边累加,降低时间复杂度
            lens=MAX(lens,dp[i][j]);
        }
for(int i=lens-1;i>=2;i--)
    square[i]+=square[i+1]; //将包含的情况累加上
```

可以大大简化时间复杂度.

3.3.5: A Game 游戏:这是一道博弈问题,状态很少,可以使用 DP 解决方程如下面所示的 $F[i][j]=\max\{\text{num}[i]+S[i+1][j]-F[i+1][j],\text{num}[j]+S[i][j-1]-F[i][j-1]\}$,因为当你在 i 到 j 中取了 i 之后,那么你就只能在 i + 1 到 j 中作为后手取值,故你所能取得的值为 i + 1 到 j 的和减去作为先手所取得的最大值;这样的思想尤为精妙,但是分析一下,之所以可以动态规划,第一它具备最优子结构,第二它拥有重叠子问题,因为序列的选择是有序的,符合规律.选要注意的是实现时需要从小到大计算.

◎第四节:

这一节的题目突然拔高了,不是说有多么难,而是题目的分析要求加大,算是和第四章衔接,主要涉及了计算几何的算法,分治策略和一个不太容易的 DP,只要多加分析,理解题意,想透彻了,问题也就迎刃而解了.

3.4.1: Closed Fences闭合的栅栏:第一道计算几何问题,令我头疼了很长时间,我的方法是极限2分法,在精度足够的情况下,进行极限2分,同时要有特殊判断,也就是是否被点挡住,当时想到了,但是写错了,应该是与自己的中位线进行对比,而不是所有的.要用的叉积,是一道不错的考察分析的题目.

```
inline double crossproduct(Point start,Point U,Point V){//计算叉积
    double Ux=U.x-start.x,Uy=U.y-start.y,Vx=V.x-start.x,Vy=V.y-start.y;
    return (Ux*Vy-Uy*Vx);
}
inline int cancross(Line A,Line B){
    double
    A1=crossproduct(A.start,A.end,B.start),A2=crossproduct(A.start,A.end,B.end),
    B1=crossproduct(B.start,B.end,A.start),B2=crossproduct(B.start,B.end,A.end);
    if(A1==0 || A2==0) return -1;//被点挡住,A为中位线时,以A为轴,有0就表明被点挡住了.
    if(A1*A2<0 && B2*B1<0) return 1;//严格相交
    return 0;
}
```

3.4.2: American Heritage 美国血统:这道题目不难,但是很经典,需要树结构的知识,要知道哪种遍历可以确定唯一的树,然后进行分治就行了.用到“下放”的思想.

3.4.3: Electric Fences 电网:pick 定理说明了其面积 A 和内部格点数目 i 、边上格点数目 b 的关系: $A = i + b/2 - 1$. 这样只要求出边上的点就行了,可以算方程,或者:一条直线 $((0,0),(n,m))$ 上的格点数等于 n 与 m 的最大公约数+1. 即 $b = \gcd(n,m) + 1$. $\gcd(n,m)$ 为 n 与 m 的最大公约数. 这样计算的时间复杂度就很低了.

3.4.4: Raucous Rockers “破锣摇滚”乐队:本章的最后一道题目,每章的最后一道题目都会让人卡一下,这是一道另类的多维背包,我的方法是转化思想:混合型动态规划:假设我们已经有一个 $reg[i][j]$ 代表从 i 到 j 的可行最大数量,那么我们对于整个序列来说只要算出 $dp[i][j]$ 代表前 i 个背包选前 j 个曲子的最大值,就相当于把序列分成 M 段,取最大值,类似石子归并类的划分问题.因此我们先用 01 背包计算出 $reg[i][j]$ 的那个区间的最优值.然后使用方程: $DP[i][j] = \max\{DP[i-1][k] + reg[k+1][j]\}$ 计算, $DP[m][n]$ 即为所求.

USACO 官方的题解很不错: $dp[a][b][c]$ 表示前 a 个包,已经录了 b 分钟,录到 c 个曲子的最大值.因为我们一旦可以在当前 a 个包内放得下曲子就没理由放到第 $(a+1)$ 个包中,同样一旦放不下就只能放到 $(a+1)$ 个包中了.

★章末总结:

第三章断断续续的花了半年通关了,感觉积累的更多了,同时本章最大的收获就是如何分析更加巧妙的算法,本章的题目我做的比前面的花的时间都长,因为的确分析的难度大了,这是其一.其二第三章仍然是基本功,第一章带着你进来,第二章稍微加强但是让你够得更高,第三章则是在此基础上真正让你学会分析的方法,问题只要分析的好总会简单的,有时候多加分析真的可以得到很大的锻炼,相信第三章是一个转折点.我得到的最大收获就是磨砺.这样我才能拥有好的解题技术,也正是前两章分析问题的基础,才能在本章不受太大的阻碍,而本章的历练也会使得在今后的训练中走的更长远.

2010/2/9