acm-template

langman

February 23, 2018

Contents

1　头文件

2 图论

2.1 二分图

判断是否二分图 求出最大匹配数

2.2 并查集

2.3 最短路

两种算法 但是要注意dijkstra无法处理负边的情况

2.3.1 dijkstra

需要注意的在于 可以更优化 我没写了 而且需要注意重边的情况

2.3.2 spfa

需要注意的是怎么建边 双向边?

2.3.3 Flody

这个就不写了,一个小dp

2.4 最小生成树

这是个什么玩意呢 图里面是吧,找到n-1条边使得生成一颗树,然后他的边权之和最小

2.5 最大流

2.5.1 Dinic

板子先存着,坑定用的着

3 数学方面

3.1 三个特别的数

3.1.1 Fib 数列

$$f(x) = f(x-1) + f(x-2)$$
$$f(0) = 0, f(1) = 1$$

3.1.2 卡特兰 数

$$\sum_{i=1}^{n} f_i * f_{n-i} = f_n$$
$$h(n) = C_{2n}^{n} - C_{2n-1}^{n}$$

注意它这个数字来自于什么情况。

3.1.3 斯大林公式

$$\sqrt{2 * PI * n} * (\frac{n}{e})^n = n!$$

3.2 数论

第一个自然是最基础的欧几里得算法，欧几里得算法的用处有很多，求最大公倍数，解方程，很多。在后面的过程会把一些常见的板子列出来，一般来说这些板子 都已经经过验证，但是不好说对吧。简单题我们可以通过一些模板直接得出答案，但是怎么说，这些对于难题 估计只能算工具，重要的是如何转换。

3.2.1 欧几里得

然后是基于这个定理得出的一个定理，中国剩余定理

3.2.2 乘法逆元

思想是通过扩展欧几里得来得出，如果缘分到了，那么 还能用费马小定理来解

3.2.3 欧拉函数

这个东西好呀，他求的是比n小的，并且和n互质的数的个数 更多的来说我觉得这个东西是一个工具，他对解一些题有很重要的作用，起到一个工具的作用 我目前学的比较浅，对他的优化作用没有很深的了解。

3.2.4 莫比乌斯函数

$$F_n = \sum_{i=1}^{n} f_i$$

$$f_i = \sum_{d|n} u(d) * f(\frac{d}{n})$$

和欧拉函数一样很重要的一个函数他的定义我就不说了，毕竟我latex学的还不好，公式的 插入对我来说用处不大。

3.3 博弈

3.3.1 主要的解题思想

官方说的是通过必败点和必胜点来判定 先通过必败点来推，直接来看必胜点，把问题抽象成图 把状态抽象成点，必败点就是先手必败点，然后通过必败点能走到的搞成必胜点，如过有一个状态没有走过 而且他后面的路都是必胜点那么他就是必败点。感觉就像dp一样，记忆化搜索。 当然题目不可能出的那么简单的。 不过根据雄爷定理，万事不离期宗，掌握基本，扩展自己去发掘。

3.3.2 题型

巴什博弈
这个是最简单的博弈，就是一堆东西，每个人自己能拿1-n件，谁最后一个拿完谁赢，这个是最简单的，不记录。
威佐夫博弈
有两堆各若干个物品，两个人轮流从某一堆或同时从两堆中取同样多的物品，规定每次至少取一个，多者不限，最后取光者得胜。 这个的解题思路在于通过前面的那个np问题来解决，用局势来思考这些问题，前几个局势在于(0,0),(1,2),(3,5),(4,7).....然后一些大佬就总结出了一些牛逼的结论$(a_k, b_k), a_k = \frac{k*(\sqrt{5}+1)}{2}, b_k = a_k + k$人才。
Fibonacci
有一堆个数为n的石子，游戏双方轮流取石子，满足：
（1）先手不能在第一次把所有的石子取完；
（2）之后每次可以取的石子数介于1到对手刚取的石子数的2倍之间（包含1和对手刚取的石子数的2倍）。 约定取走最后一个石子的人为赢家。 结论是 当n为Fibonacci数时，先手必败
尼姆博弈
有三堆各若干个物品，两个人轮流从某一堆取任意多的物品，规定每次至少取一个，多者不限，最后取光者得胜。 这个博弈有点意思 他的必败点的局势在于$(a, b, c) a \wedge b \wedge c = 0$

3.3.3 SG函数

这个在看之前感觉很高级但是啊，好像也就是一个dp的过程，通过一个必败点，看成起点然后，那个方法看成通向下一个起点的路，然后找所有能直接到这个必败点的必胜点。好像也就那么回事。好像能解决的都是小数字题这是一个板子，f里面存的是方法，多堆问题可以转化成异或来解决。

3.3.4 解题策略

* 1 ： 相信自己的第一感觉
2 ： 博弈都会和一些特别的数搭边 ， 所以第一件事坑定是分析局势然后找找看是不是有特别的意义，像什么 卡特兰数， f i b 数列 ，幂次方，异

或的值是否为 0 ;
3 : 不挂怎么说，记得打表。

3.4 组合数学

3.4.1 求组合数

第一个是求组合数,方法很多不去列举，注意的是一般来说，组合数都是需要去 模一个数，所以他的分母在计算的时候是需要去求逆元的

3.4.2 polay定理

设G=p1，p2，. . . ，pt是X=a1,a2,. . . ,an上一个置换群，用m种颜色对X中的元素进行涂色，那么不同的涂色方案数为

$$\frac{1}{G}\sum_{k=1}^{t} m^{Cyc(p_k)}$$

$Cyc(p_k)$是置换p_k的循环节个数

3.4.3 lucas定理

当组合数的基数过大的时候进行这些操作但是注意，我们的操作也是要求那个模数为素数，且 模数要小的情况下，素数的情况我们可以用扩展lucas定理来解决。一个工具，一个数论上的分支。

3.5 线代

3.5.1 矩阵快速幂

这类方法，很多是用在递推关系式的时候，像什么fib数列什么的。

3.5.2 矩阵方面知识

就是用高斯消元法去解决一些问题，像什么秩和方阵的值。

3.6 计算几何

3.6.1 几何基本知识

矢量
矢量的乘积有很多的作用，注意定义。 适用点在于：1:面积 2：位置
跨立实验与判断两线段是否相交
线段P_1P_2, Q_1Q_2,相交的条件为
P_1Q_1xP_1P_2 * P_1P_2xP_1Q_2 >= 0
Q_1P_1xQ_1Q_2 * Q_1Q_2xQ_1P_2 >= 0

*pick*定理
线段上的是整数点的数的个数 求gcd;
PICK定理 设以整数点为顶点的多边形的面积为S，多边形内部的整数点数为N，多边形边界上的整数点数为L，则 S=L/2 + N-1

3.6.2 判断点是否在多边形中

为1的时候，则在内部。2，应该是边上。

3.6.3 凸包问题

4 状态转移 dp

dp的定义： 1：记忆化搜索；2：状态转移 所以我们的解决方案总是跟着这个来走，从定义出发。 难点集中于两个方面，状态式的确定和状态转移方程的确定

4.1 背包

背包的问题主要以下几种： 01背包，部分背包，完全背包;[相对来说比较简单] ，分组背包[个人感觉较难] 背包难在如何， 确定维数，确定背包的容量是什么以及背包的价值是什么，还有背包的dp关系转移式。

4.2 树形dp

关键点在于找状态点间的关系，他一般只有三个关系，父亲节点 ，儿子节点，还有兄弟节点，去找他们之间的关系，所以一般是两遍dfs 找父亲与儿子的关系，找儿子与父亲的关系。

4.3 数位dp

这个dp的精髓在于记忆化搜索，也就是在最高位不是被限定的情况下进行记录，这样的话省掉很多多余的步骤。 所有的出发点都处于这个目的。

4.4 状压dp

这个dp的精髓在于状态转移，不过能压缩的情况也是很限定的。 像什么每个点的状态在于都是能用两个状态来描述，且这些点不多，但是组合的方式很多。 一些状压dp经常用的上的公式。

4.5 一些常见的dp

4.5.1 LIS 最长上升子序列