矩阵快速幂，状压dp，数位dp，树形dp

北京理工大学软件学院
科技创新基地
算法实验室
贾容千

矩阵快速幂

- 1.快速幂
- 2.矩阵与dp转移方程
- 3.例题

快速幂

- 快速幂的思想：
- 假设我们要求a^b，最朴素的方法就是不断地乘a，乘b次，复杂度O(b)。
- 如果b很大，10^9，就需要用快速幂的思想。
- 例：a=3，b=100；
- 100的二进制为：1100100
- 也就是100可以化成64+32+4。
- 所以原数可以化成a^64*a^32*a^4

快速幂

- 算法流程：
- 判断1100100的每一位是否为1，如果是1，就乘对应的二进制次幂。以此类推，直到乘完全部的位数。
- 时间复杂度O(log n)

快速幂

- 代码：
- int quickpow(int a,int b){
- ans=1;
- while(b){
- if(b&1) ans*=a;
- a*=a;
- b>>=1;
- }
- return ans;
- }

矩阵与dp

- a[i]=a[i-1]+b[i-1]+1,b[i]=2*a[i-1]-5 ;a[1]=1,b[1]=1,问a[x]=?,b[x]=?
- 很简单的递推，一步步推即可，但是，如果x是10^9，如何推？
- 思维：递推式可以化为矩阵乘积

$$[\ a[i]\ ,\ b[i]\ ,\ 1\ ,\ 5\ ] = [\ a[i-1]\ ,\ b[i-1]\ ,\ 1\ ,\ 5\ ] * \begin{bmatrix} 1 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

矩阵与dp

- 那么，矩阵A[i]=A[i-1]*B;
- A[i+1]=A[i]*B=A[i-1]*B*B
- A[x]=A[1]*B*B*B.....=A[1]*(B^(x-1));
- 因为矩阵乘积可以换乘积顺序，所以可以先算出B^(x-1)，如何计算呢？
- 快速幂！

问题迎刃而解~

- 1.构造出递推矩阵
- 2.对构造出的矩阵B，进行B^x的快速幂，乘积换成矩阵乘法。
- 3.最后矩阵的第一行第一列和第二列就是a[x]和a[y]。

例题

- 2015多校联合赛#3 1003（hdu 5318）
- 题意：
- 给定n个字符串，如果一个串的尾和另一个串的头重合，则这两个串可以拼接。问：公选m个串，总共有多少种拼接方法？(n<=50,m<=1e9)

dp状态

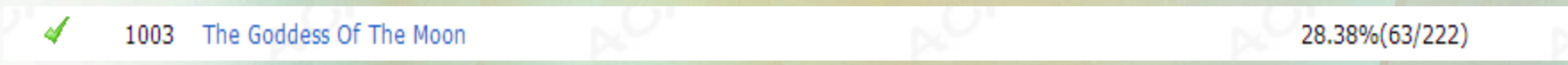
- 分析：
- 对于已经选的串，前面多少种拼接和之后无关，后面可以拼什么只和最后一个串有关。
- 设dp[i][j]表示以i串为结尾，共接了j个串的方法数。
- 那么dp[i][j]=dp[k][j-1] (k为所有可以接在i之后的串)

dp状态

- for(i=1;i<=m;i++){
- for(j=1;j<=n;j++){
- for(l=1;l<=n;l++){
- if(judge(j,l)) dp[j][i]+=dp[l][i-1];
- }
- }
- }
- 复杂度为O(n*n*m),m<=1e9.....

构造矩阵

- A[i]=[dp[1][i],dp[2][i],dp[3][i],.....dp[n][i]]
- 如果i，j可拼接，B[i,j]=1，否则B[i,j]=0。
- A[m]=A[1] * (B ^ (m-1));
- 矩阵快速幂即可~
- 答案为A[m]里的各项之和~
- 比赛中的代码大家可以参考~



状态压缩dp

- 1.记录两种状态的dp方程
- 2.二进制的引入

从那场codeforces说起

Codeforces Round #321 (Div. 2)

Final standings

You may double click into cells (or ctrl+click) to view the submissions history or hack the solution

Standings

#	Who	=	*	A 750	B 1250	C 1500	D 2000	E 2500
415	xinging	3060		726 00:08	1110 00:28	1224 00:46		
535	fange121	2917		679 00:07	1130 00:24	1108 00:57		
1875	paul-lu	735		735 00:05	-11			
2670	RONALDOsss	679		679 00:07	-8			
3006	1120141951	622		622 00:26	-4			
	Accepted Tried			3460 3897	1543 2976	1310 1978	309 598	4 51

- problem D没人做出来？
- 学会状态压缩dp，你就会啦

codeforces#321 div2 D

- 有n(n<=18)个景点，每个景点有满意值a[i]。k个关系，每个关系有x,y,c，意为如果连续去x和y，将会有满意值c。
- 问：选其中m个景点，如何使得满意值最大？

dp方程

- 我们发现，走过景点的集合会影响后面的走法，而走过景点的顺序不会影响后面的走法。
- 那么，假设我走过1,2,3,4,5,且停在节点3，那么1,4,5,2,3还是1,5,4,2,3都是不影响之后的路线的。假设我定义et表示已经走过1,2,3,4,5。st表示走过1,2,4,5。
- dp[i][j]表示走过的点为i情况，目前停留在j点所能获得的最大能量。
- dp[et][3]=max(dp[st][2]+work(2,3))+a[3];

引入二进制

- 对于每个景点，我们都有“去过”和“没去过”两种选择。
- 总共景点数只有18个。
- 那么总共的情况数为2^18。
- 我们可以分别用2^18个整数来代表所有的状态。如何分配这些整数去代表某种状态呢？
- 二进制！

引入二进制

- 例如：景点数为5个，那么我们用0-31这2^5个数来表示所有状态，这些数恰好二进制都有5位，那么我们可以让每一位对应一个景点。如果状态数字为14：
- 14的二进制为：01110。
- 就用它表示走过2,3,4号景点，没去过1,5号。
- 如果是23（10111），就表示去过1,2,3,5号景点，没去过4号景点。
- 这样的表示方法不会重复也不会遗漏。
- 比赛代码可以参考~
- <<，>>，&，|，^ 位运算符号一定要掌握~

数位dp

- 小俊很喜欢数学，现在他要给你出一道严肃的数学题。
- 定义 F[x]F[x] 为x在十进制表示下各位数字的异或和，例如F(1234) = 1^2^3^4=4。给你两个数a,b(a≤b)。
- 求F[a] + F[a+1] + F[a+2]+…+ F[b−2] + F[b−1] + F[b] 模 10^9+7的值 。

dp状态

- 我们把问题转化成一个按位取数的过程。
- 我们发现，0-9所有数的抑或值范围应该在0-15之间。
- 对于前面已经选完的若干个数，我们不在乎选了谁，只在乎这些数抑或和是几。
- 定义dp[i][j]表示选了i位，抑或和为j的方法数。
- dp[i][j]+=dp[i-1][l]^(k); (l^k=j)
- 每个抑或值乘以方案数即可。

类似取数问题

- 定义好数为
- 1.每一位乘积模7等于6
- 2.每一位都可以被上一位整除
- 3.奇数位之和等于偶数位之和
- .....
- 问1-n的所有数中，有多少个“好数”？

树形dp

- 1.递归
- 2.树上的dp方程

递归

- 深入了解递归的机制。
- void dfs(int x){
- 	if(x>3) return;
- 	printf("A(%d)",x);
- 	dfs(x+1);
- 	printf("B(%d)",x);
- }
- dfs(1),输出？
- A(1)A(2)A(3)B(3)B(2)B(1)
- 练习codeforces#321 div2 C

树形dp

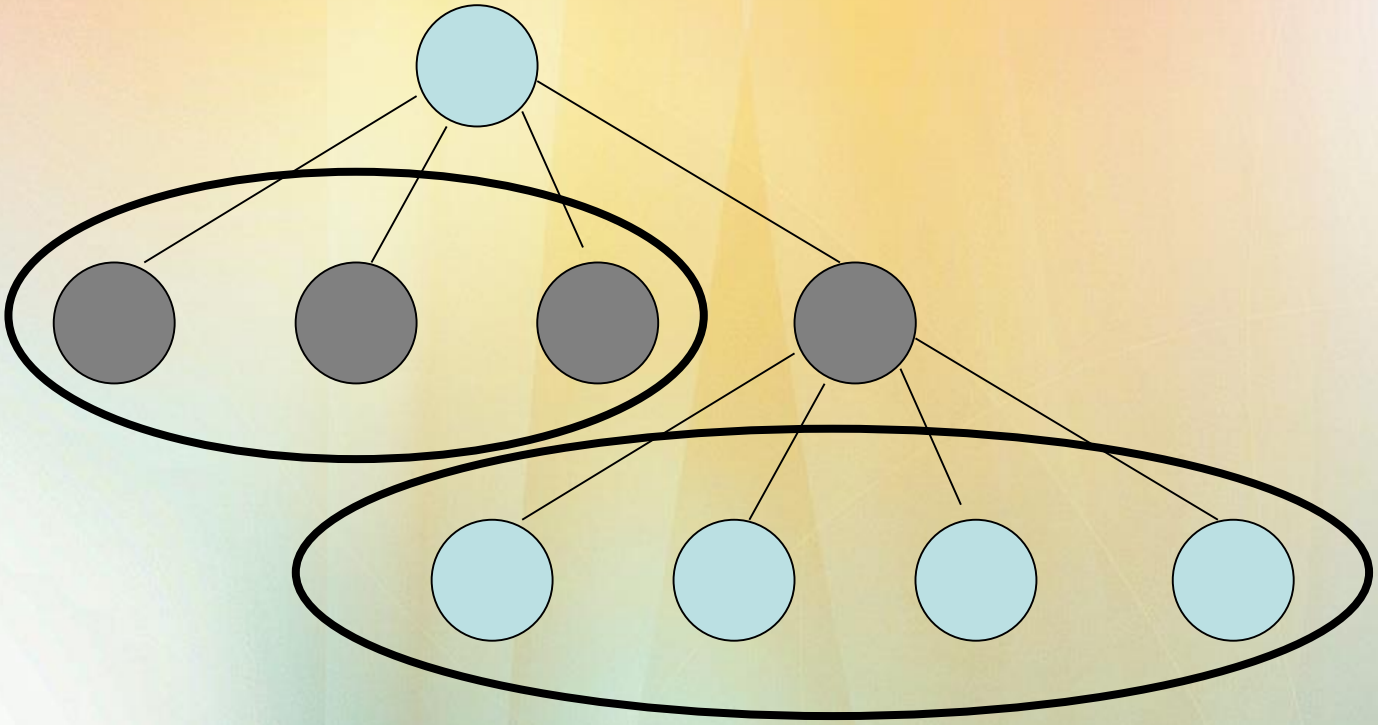
- 树形+dp
- 在一颗树上进行的dp，要求对dfs有很深刻的了解。
- 往往是根据儿子节点的最优解递推出父亲节点的最优解，由底向上更新。
- 由于父亲节点的dp值需要儿子节点更新，所以往往是先搜索，后更新。
- dfs(){
- for(){//遍历儿子节点
- dfs();//搜索子节点
- for() dp[]=dp[]+...;//更新dp值
- }
- }

树形dp

- 题目大意：
- n个人形成一个关系树，每个节点代表一个人，节点的根表示这个人的唯一的直接上司，只有根没有上司。要求选取一部分人出来，使得每2个人之间不能有直接的上下级的关系，求最多能选多少个人出来?
- 这是一个经典的树型动态规划。
- 状态？
- 转移？

树形dp

- 简单的染色统计是不正确的



树形dp

- 人之间的关系形成树型结构

- DP, 用dp[i][0]表示不选择i点时，i点及其子树能选出的最多人数，dp[i][1]表示选择i点时，i点及其子树的最多人数。

树形dp

- 状态转移方程:
  - 对于叶子节点k， dp[k][0] = 0, dp[k][1] = 1
  - 对于非叶子节点i,
  - dp[i][0] = ∑max(dp[j][0], dp[j][1]) (j是i的儿子)
  - dp[i][1] = 1 + ∑dp[j][0] (j是i的儿子)
- 最多人数即为max(dp[0][0], dp[0][1])