**[对一些DP题目的小结](http://www.cppblog.com/qywyh/archive/2007/04/20/22399.html)**

PKU 3093 Margaritas on the River Walk  
        先对输入的数组排序，然后类似于01对a[i]做决策，核心代码加了注释：  
         for (i=1; i<=n; i++) {  
                 for (j=1; j<=maxsum; j++) {  
                        if (j >= sum[i]) d[i][j] = 1; //j比sum[i]大，肯定这时候d[i][j]=1;  
                        else {  
                                d[i][j] = d[i-1][j];//不考虑a[i]  
                                if (j-a[i]>=0) {//考虑a[i]  
                                         if (d[i-1][j-a[i]] > 0) d[i][j] += d[i-1][j-a[i]];//把a[i]加进以前的选择里面  
                                         else d[i][j]++;//a[i]单独作为一个选择(这里需要先对a[i]排序，消除后效性)  
                               }  
                        }  
                 }  
         }   
  
PKU 1037 A decorative fence   
        先dp算出以i为起点的序列的个数，再组合数学  
        td[n][i]和tu[n][i]分别表示个数为n，以i开始的上升和下降的序列个数  
        易知:  
        td[n][1] = 0;   
        td[n][i] = sigma(tu[n-1][j], j从1..i-1)  = td[n][i-1] + tu[n-1][i-1] ;  
        tu[n][i]  = td[n][n+i-1];  
  
PKU 2677 Tour  
        双调欧几里德旅行商问题(明显阶段dp)  
        动态规划方程 ：d[i+1][i] = mint(d[i+1][i], d[i][j]+g[j][i+1]);   
                                      d[i+1][j] = mint(d[i+1][j], d[i][j]+g[i][i+1]);  
                                       0<=j<i

PKU 2288 Islands and Bridges  
        集合DP  
        状态表示: d[i][j][k] (i为13为二进制表示点的状态, j为当前节点, k为到达j的前驱节点)

pku 1014   已做  
pku 1037     
pku 1050   已做  
pku 1088   已做  
pku 1141   已做  
pku 1159   已做  
pku 1163   已做  
pku 1322   AC  
                  看到题目就害怕，概率的-\_-结果分析之下原来也不难  
                  状态d[i][j]表示有j种颜色,拿了i个巧克力的最优值  
                  方程: d[i+1][j+1] = d[i][j]\*(c-j)/c;               (c为总颜色数)  
                            d[i+1][j-1] = d[i][j]\*j/c;  
                  由于只是保留3位小数，所以加优化if (n>1000) n = 1000+n%2; //至于为什么要分奇偶性，这个还不太懂-\_-这道算是ac一半而已  
pku 2904   AC  
                  dp[k][i][j]表示k个邮筒时候放鞭炮数为i..j时候的最优值  
                  转移方程为:  
                  dp[k][i][j] = min{t+max(d[k-1][i][t-1],d[k][t+1][j])};  
                  状态转移时候就是考虑选t个鞭炮放时候爆或不爆  
pku 1458   已做  
pku 1579   已做   
pku 1695   AC   
                 d[i][j][k]表示到达第i个点时候另外两辆车分别在点j和k时候的最优值  
                  方程： d[i+1][j][k] = min(d[i+1][j][k], d[i][j][k]+g[i][i+1]);  
                               d[i+1][i][k] = min(d[i+1][i][k], d[i][j][k]+g[j][i+1]);  
                               d[i+1][i][j] = min(d[i+1][i][j], d[i][j][k]+g[k][i+1]);  
                  //初始条件d[1][1][1] = 0;  
pku 1732   AC  
                  线型模型，本想用trie的，结果用map偷懒了。  
                  d[i] = min{d[j]} + 1      0<=j<i && j+1..i字符合法  
pku 1953   已做  
pku 1976   AC  
                  先对区间做预处理, 后面不足的coaches补0;  
                  d[k][j] = max{d[k-1][p]}+b[j];          0<=p<=j-m (b为处理后的区间数组,m是一台locomotiv的容量)  
                  由单调性可以在状态转移时候保存前一次转移时候的最大值再和b[j-m]做比较，把O(n^2)压缩到O(n)的时间复杂度  
pku 2386   已做  
pku 2479   已做  
pku 2951   已做        
pku 3036   已做  
pku 3014   已做  
pku 2229   已做  
pku 1185   AC  
                  最经典的状态DP,我用三进制表示每行状态，然后递推，结果tle,分析之后，枚举出有效状态，再推, 1000ms左右，  
                  还是不够 快， 张伟达的论文上有更快的算法。  
pku 1276   AC  
                  01背包  
  
有空把以前的也再做一次!~