状压与区间

2019年1月26日

黄哲威 hzwer
北京大学16级计算机科学

第四节目标

- 掌握简单的区间动态规划
- 认识用二进制表示状态
- 掌握简单的状态压缩动态规划
- 本节课件参考北京大学叶伽宁，清华大学王逸凡同学课件

区间 dp

石子合并

- 有n堆石子排成一列，每堆石子有一个重量w[i]。
- 每次合并可以合并相邻的两堆石子，一次合并的代价为两堆石子的重量和w[i]+w[i+1]。
- 问安排怎样的合并顺序，能够使得总合并代价达到最小。
- n<=100。
- CodeVS 1048

石子合并

- 状态：f[i][j]表示把第i堆到第j堆的石子合并在一起的最优值。
- 转移：考虑到i~j必然是由两个子段合并而来。
-         枚举k，i~j由i~k和k+1~j合并而来。
-         f[i][k]+f[k+1][j]+(sum[j]-sum[i-1])
-         ->以此更新f[i][j]，取min；

石子合并

初始化f[i][i]=a[i],其余为正无穷。

```
for(int i=1;i<=n;i++)
    for(int j=i+1;j<=n;j++)
    for(int k=i;k<j;k++){
        int now=f[i][k]+f[k+1][j]+sum[j]-sum[i-1];
        f[i][j]=min(f[i][j],now);
}
```

答案为f[1][n]。

区间 dp

- 序列: 区间[l,r]的 最后一步处理的是k元素 / 分割点是k / 受k控制的

- 环: 复制一下粘到后面, 枚举断点

区间 dp

- 序列: 区间[l,r]的 最后一步处理的是k元素 / 分割点是k / 受k控制的
- 环: 复制一下粘到后面, 枚举断点
- 区间 DP 的题目的转移和普通 DP 不太一样。正常的 DP 中，如果设状态为 f[i, j, k]，一般是先按顺序枚举 i，再枚举 j，最后 k。区间 DP 的特点是，第一个枚举的量是区间长度，转移是从区间长度由短到长进行的。剩下的部分与普通 DP 一致。
- 在思考区间 DP 题时，一个很常见的思路是枚举当前区间内的某个位置，把这个位置作为这个区间最先处理/最后处理的元素，然后考虑在这个位置把区间切开(也有可能直接枚举切开区间的位置)， 把当前区间的问题转化成两个小区间内的子问题。根据这一步子问题的化归来写方程。

乘积最大

- 给定长度为 n 的一个数字串。
- 你需要在串中间插入 k 个乘号，使得构成的表达式的值最大。
- $n \le 40$，$k \le 6$。

乘积最大

- 一道非常经典的区间 DP 题。
- 设 f[i,j,k] 表示在 i 到 j 这个区间插入 k 个乘号所能达到的最大乘积。
- 枚举 l，表示 i 到 j 这个区间中的第一个乘号放在第 l 个数后面。
- 则 f[i, j, k] = max{g[i][l] * f[l+1][j][k-1]，其中 $g_{i,l}$ 表示 i 到 l 这个区间构成 的数。

能量项链

- 有 N 颗有头标记与尾标记的珠子串成的一串。对于相邻的两颗珠子，前一颗珠子的尾标记一定等于后一颗珠子的头标记。

- 如果前一颗能量珠的头标记为 m，尾标记为 r，后一颗能量珠的头标记为 r，尾标记为 n，则聚合后释放的能量为 m*r*n (Mars 单位)，新产生的珠子的头标记为 m，尾标记为 n。 请你设计一个聚合顺序，使一串项链释放出的总能量最大。

- $1 \leq N \leq 100$

能量项链

- 明白题意以后，不考虑环的情况，就是个简单的区间dp
- $F(i,j) = F(i,k) + F(k+1,j) + a[i] * a[k] * a[j]$
- 枚举环的断点或者复制一遍环枚举起点

状压 dp

状压DP

- 状压DP就是状态表示比较复杂的一种DP。
- 不像序列DP、树形DP之类的直接dp[i]表示i这个前缀，或者i这个子树的相关信息。
- 状压DP要求表示的状态往往是一个集合的一个子集、一张图的一个子图，或者其他什么乱七八糟的东西，需要你手动把这些状态和自然数之间建立起一个一一对应关系方便写代码而已。

状压DP

- 由于NOIP中几乎都是子集DP，也即有一个集合S={0,1,...,n-1}，一个状态是S的一个子集。这时一般用一个二进制数x来表示S的一个子集。如果x对应位上是1就表示该数在子集内，否则不在。
- 要对于这种集合进行操作，因此位运算必须熟悉的。
- 下面回顾一下位运算的操作：

位运算

- &：取交集
- |：取并集
- ~：取补集
- ^：对称差
- <<,>>：左移右移（做一些特殊的操作）
- x>>i&1：取出第i位

匹配

- 给一张二分图，左边n个点，右边m个点，问有多少个匹配使得左边的每个点都在匹配中。
- 答案对10^9+7取模
- n<=15,m<=100

匹配

- 记dp[i][S]表示考虑了右边前i个点的匹配情况，匹配了左边S这个子集的点，总方案数。
- dp[i][S]→dp[i+1][S]
- dp[i][S]→dp[i+1][S∪{k}]　　(k,i+1) $\in$ E && k$\notin$ S
- 复杂度O(m*2^n)

匹配

```
readln(n,m);
for i:=1 to n do
begin
    read(k);
    for j:=1 to k do
    begin
        read(x);
        a[i][x]:=1;
    end;
end;
for i:=1 to m do
begin
    f[i][0]:=1;
    for j:=1 to (1 << n)-1 do
        for k:=1 to n do
            if (((j >> (k-1)) and 1)=1)and(a[k][i]=1)then
                f[i][j]:=f[i][j]+f[i][j xor (1 << (k-1))];
end;
writeln(f[m][(1 << n)-1]);
```

SP

- 给一张有向带权图，求最短的Hamilton路径。
- n<=15
- Hamilton路径是指从某个点开始经过每个点恰好一次的路径。

SP

- 设dp[S][i]表示路径以i结束，经过点的集合为S，此时的最短路。
- 枚举下一个点j（j∉S），转移到dp[S∪{j}][j]即可。
- 复杂度O(2^nn^2)

互不侵犯KING

- n*n的board里放置国王，使得它们互不能攻击。求方案数模10^9+7。
- 一个国王能攻击到它周围8个格子。
- n<=9

互不侵犯KING

- 记dp[i][S]表示确定了前i行，其中第i行S这个子集放了国王。的方案数。

- 枚举下一行的状态，判断是否有互相侵犯。这可以用位运算O(1)实现。

- 复杂度O(n*2^(2n))

排列PERM

- 给一个数字串s和正整数d，统计s有多少种不同的排列能被d整除（可以有前导0）。

- 例如123434有90种排列能被2整除，其中末位为2的有30种，末位为4的有60种。

- |s|<=10,d<=1000

互不侵犯KING

```
function get(x,y:longint):bool;
begin
    if (x and y)<>0 then exit(false);
    if (x and (y >> 1))<>0 then exit(false);
    if (x and (y << 1))<>0 then exit(false);
    if (x and (x >> 1))<>0 then exit(false);
    if (y and (y >> 1))<>0 then exit(false);
    exit(true);
end;
begin
    readln(n);
    f[0][0]:=1;
    for i:=1 to n do
        for j:=0 to (1 << n)-1 do
            for k:=0 to (1 << n)-1 do
                if(get(j,k))f[i][j]:=(f[i][j]+f[i-1][k])mod 1000000007;
    for j:=0 to (1 << n)-1 do
        ans:=(ans+f[i][j])mod 1000000007;
    writeln(ans);
end.
```

装背包

- n个物品每个体积w[i]，m个背包每个大小a[i]。
- 问将所有物品装入背包最少需要多少个背包。
- Task1：n<=13,m<=100
- Task2：n<=18,m<=100

装背包

- 首先一定是挑选容积最大的几个背包来装，而且最多用n个，因此保留最大的n个背包即可。
- dp[S]表示最小的K，使得最大的K个包能够装好S这个集合。
- 枚举第dp[S]+1大的那个包装了哪些物品，进行转移。
- 复杂度O(n*3^n)

装背包

- 其实可以dp[S] = (n, v)记一个pair，表示至少要几个包，以及最后一个包最多能剩多少空间。
- 这样就不用枚举子集了，只要枚举单件物品。复杂度O(2^n*n)。

连通图

- 问一张无向图生成连通子图的个数。答案对10^9+7取模。
- n<=16

连通图

设dp[S]表示S这个集合的生成联通子图数量。

初始化dp[S]为S的生成子图个数，即2^cnt(S)，其中cnt(S)为S内部的边数。

然后减掉不连通的情况。

随便固定S中一个点u为特殊点（比如S中编号最小的点）

然后枚举u所在的联通块u∈T \subset S，dp[S]-=dp[T]*2^cnt(S-T)。

复杂度O(3^n)

Q & A