



2004北京赛区试题分析

SZ

Finding Nemo (PKU2049)

问题简述

平面上有个迷宫，迷宫里有 M 堵墙，墙是无法通过的。还有 N 扇门，门是可以通过的。以坐标的形式给出墙和门的位置。假设你初始位置在 $(0,0)$ ，给出一个位置 (x,y) ，求去这个位置需要打开最小的门数。

所有坐标范围都在 $[1,199]$ 。

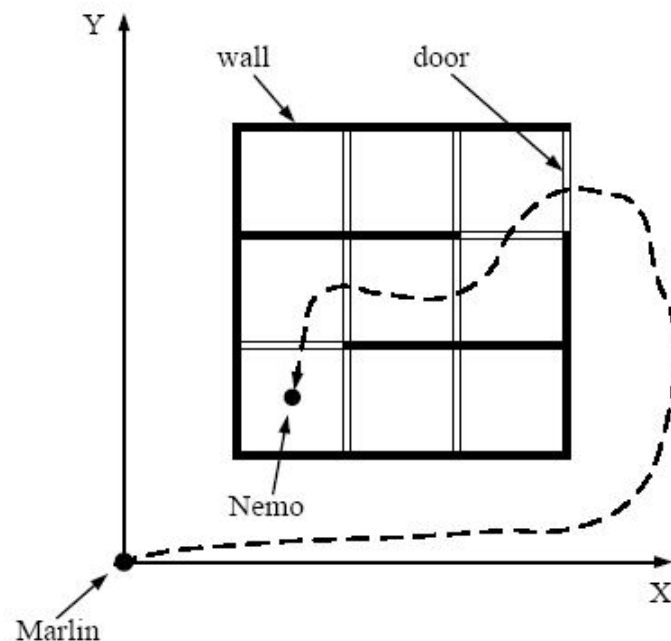
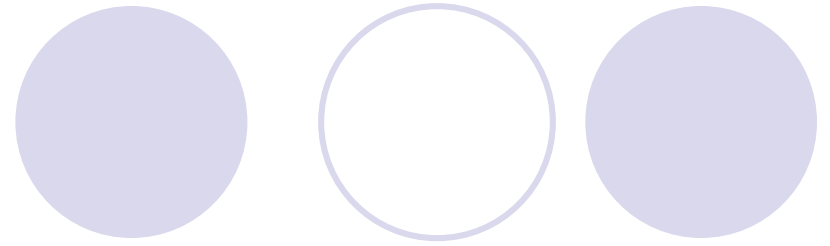


Figure-1. Labyrinth and Path

Finding Nemo



问题分析

该题就是求一个最短路径。由于题目给出的只是墙和门的坐标，所以还得预处理，划分出房间后 **Dijkstra**。或者 **BFS**。具体时限中，可以从目标点开始往回 **BFS**，代码量比划房间小些。

Searching the Web (PKU2050)

问题描述

有N篇文章，有M个请求，请求有四种格式：
一个单词A。要在N篇文章中找到关键字A，并且输出A所在的那一行。

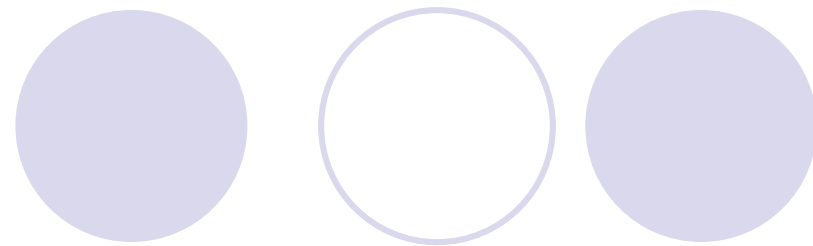
A AND B。要在N篇文章中找到关键字A和B，并且输出A，B所在的行。

A OR B。要在N篇文章找到关键字A或B，并且输出A，B所在的行。

NOT A。要在N篇文章中找到不包含关键字A的文章，并输出整篇文章。

$0 < n < 100$, $0 < M \leq 50000$ 。

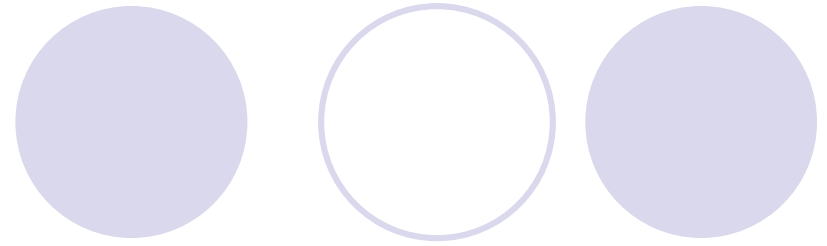
Searching the Web



问题分析

这道题中，可以建立从单词到文章的索引。建立的方法，可以使用hash，如果单词S出现在文章T中，就将(S,T)记在hash中。在C++中，也可以使用map来检索。这样对要检索的单词可以快速的找出出现过的位置。

Argus (PKU2051)



问题描述

有一些询问，每个询问有一个不同编号 q_num ，以及出现周期为 P ，假设从0时刻开始计时，问前 K 个出现的询问分别是什么。如果同一时刻有多个询问出现，按他们的 q_num 从小到大排序。

$0 \leq q_num \leq 3000$, $0 < P \leq 3000$, $K \leq 10000$ 。

Argus



问题分析

由于这题给的范围不大，可以直接按照题目描述模拟。每次顺序查找一个最小的然后输出，那么复杂度将是 $O(q_num * k)$ ，是可以接受的。更好的模拟方法是用堆，每次直接取出一个最小的，那么复杂度将是 $O(\log(q_num) * k)$ 。

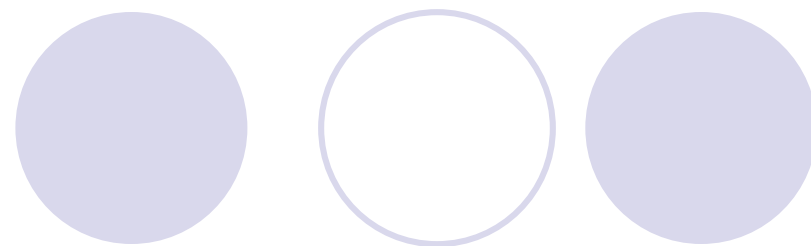
Fun Game (PKU2052)

问题简述

有几个小孩围着树玩游戏。每一轮从某个小孩开始往他左边或右边传手帕。一个小孩接到手帕后（包括第一个小孩），在手帕上写下自己的性别，男孩写（**B**），女孩写（**G**），然后按相同方向传给下一个小孩，也就是每一轮传手帕的方向经第一个小孩确定了以后就不会改变。每一轮可能在任何一个小孩那停止。现在游戏已经进行了 N 轮，已知 N 轮中每轮手帕上留下的字，求最少可能有多少个小孩。

$2 \leq N \leq 16$ 。每轮手帕上的字数不超过100个。

Fun Game



问题分析

首先可以看出，如果有一个字符串完全包含于其他某个字符串，那么这个字符串将对结果没有影响，所以先预处理去掉这些字符串。

然后先考虑所有轮传手帕都是一个方向的，且小孩排成一列。那么问题就是将 N 个字符串连成最短的字符串。如下图所示。



Fun Game

问题分析

这样只要先进行预处理，算出每对字符串连接后可以去掉的公共部分的长度。然后利用**16**位二进制表示每个字符串的两个状态：取与不取。这样可以用`opt[i][j]`来表示，当前取过的字符串由*i*表示，最后去到的字符串是*j*，可以去掉的公共部分总长。这样可以得到一个 **$16 \times 16 \times 2^{16}$** 的动态规划。

下面再考虑传递的两个方向，其实只要考虑每个字符串的反向，这样就成了 **$32 \times 16 \times 2^{16}$** 的**DP**。尚且可以接受。

Fun Game

The title 'Fun Game' is positioned on the left side of the slide. To its right, there are four circles arranged horizontally. The first circle is solid light purple. The second circle is an outline of a light purple circle. The third circle is also an outline of a light purple circle. The fourth circle is solid light purple.

最后考虑链变成圈。可以想到，每个状态需要再加一维表示以哪个字符串开头。这样最后就可以计算首尾连后去掉的公共部分。不过时间复杂度将因此再提升一个等级。

可以想到，在链变成圈的过程中。需要考虑首尾相连的情况，仅在一种情况下，也就是组成链的每一个子串都互相交叠。这个时候我们不需要考虑字符串从哪一个元素开头，任取一个作为开头对结果没有影响。

Fun Game



其它的情况下，我们通过旋转环，总可以找到一个位置断开环，那个位置没有子串的交叠。由于没有子串的交叠，我们不需要记录开头的串，这样也可以和链通过相同的复杂度算出答案。

这样，我们就得到了一个 $O(n^2 \cdot 2^n)$ 的算法。

Square (PKU2053)

问题简述

在 $[0,1] \times [0,1]$ 内，有 n 个点。这 n 个点加上 $(0,0), (0,1), (1,0), (1,1)$ 一共 $n+4$ 个点，可以将他们相连，使得所有点之间都互相连通。现在请你移动这些点，找到一种总长最小的连接方案。那么每个点移动距离和至少是多少？

$$1 \leq n \leq 100$$

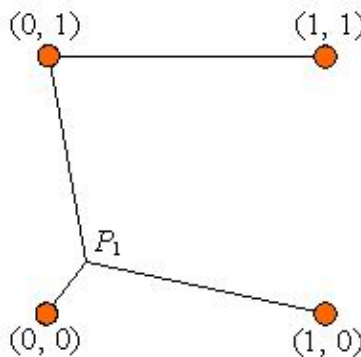


Figure-1

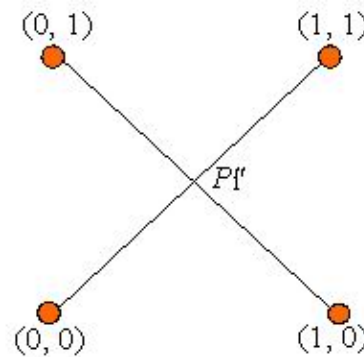
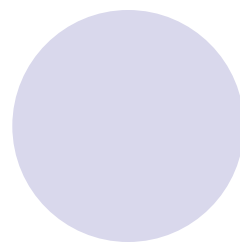
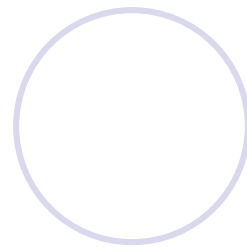
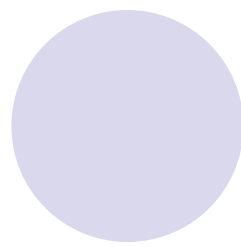
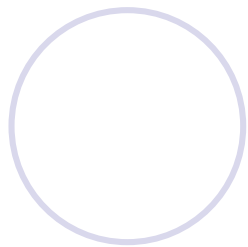


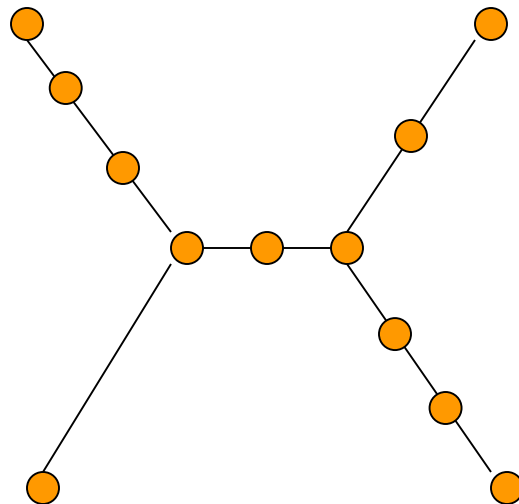
Figure-2

Square



问题分析

事实上，这题所要求的，就是移动点之后，使得最小生成树的权值达到最小。那么什么时候可以有这个最小权值呢？我们有一个结论：当 $N \geq 2$ ，最小的连法都有相同的形状：



Square

这个形状中，有两个点的度为3，位置分别是 $(\sqrt{3}/6, 0.5)$ 与 $(1-\sqrt{3}/6, 0.5)$ 。图中连接了5条线，每条线上可能分布着若干个点，每个点的度为2，不会影响总长度。这样，只需要枚举两个度为3的点，其余的点移到最近的线段上去即可。

对于1的情况非常简单，只要将此点移到 $(0.5, 0.5)$ 即可。

Color a Tree (PKU2054)

问题简述

给出一棵 N 个结点的树，每个结点有一个权值 C_i 。现在给树上色，只有当一个结点的父结点被上色后，其子节点才可以被染色。开始时只能给根结点上色。第 i 个被染色的点 V_i ，花费的值是 $i * C_{V_i}$ 。请给出最优的染色方案的花费。

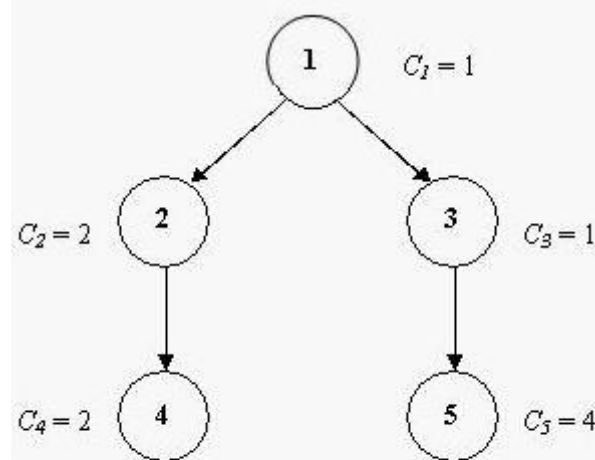
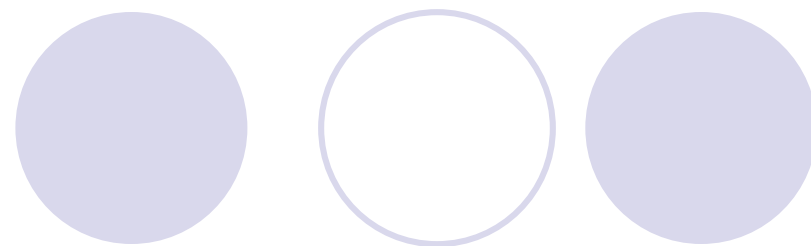


Figure-1. A tree with five nodes

Color a Tree



问题分析

这个题目其实是将结点排列为 $V_1V_2\dots V_n$ ，保证一个结点一定在它祖先之后。那么，总花费就是 $\sum C_{v_i} * i$ 。直观地看，权值越小的点应当放得越前。如果没有树结构的限制，当然按照权值从大到小的顺序来排列。那么加上树结构也是类似的。

首先，权值最大的点 V ，如果其父节点为 Fv ，那么 V 一定排在 Fv 的后面一个。也就是说，无论这两个点排在哪里，一定是相连的，所以我们可以将它们看作一个点，然后用它们的**平均值**来代替它们。这个时候，如果这两个点原本已是几个点的平均值，那么要用加权平均来替代它们。反复这个过程，直到所剩结点按照降序排列。再将合并的节点展开即是结果。

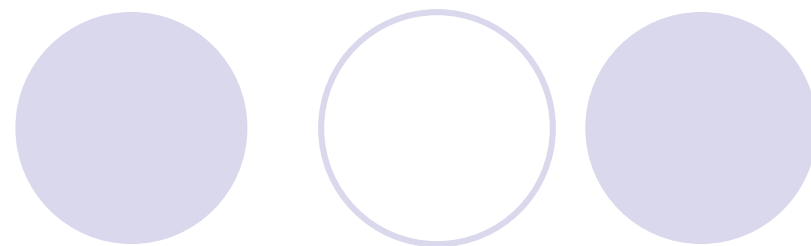
Kid's Problem (PKU2055)

问题简述

有一只密码锁，有 N 个齿轮和 N 个转盘，每个齿轮有 M 个齿。一个转盘关联着若干齿轮，当一个转盘 i 被转过一格，其关联的第 j 个齿轮将被转过 k_{ij} 格。已经知道打开密码锁时第 i 个齿轮被转过的角度 B_i 。请问至少要旋转多少次转盘？

$$1 \leq N \leq 20, \quad 1 \leq M \leq 10$$

Kid's Problem



问题分析

对于第j个齿轮被打开的条件，我们可以列出一个模线性方程

$$C_{j1}X_1 + C_{j2}X_2 + \dots + C_{jN}X_N \equiv B_j \pmod{M}$$

X_i 表示第i个转盘转过的格数， C_{ji} 表示i每转过一格j所转过的格数。如果转盘i与齿轮j没有关联，那么 $C_{ji}=0$ 。然后利用高斯消元解这个方程组就可以了。由于M并非质数，所以在解的过程中会出现一个方程有多解的情况。这时候，我们需要枚举解，回带验证。

Kid's Problem

有一个值得注意的地方是，由于是模线性方程组，在高斯消元中，使用普通的加减消元容易出现问题。原因是以下两个方程

$$Ax \equiv B$$

$$kAx \equiv kB \pmod{M} \quad (k \neq 0)$$

并不等价。这样一般的消元会产生很多增根，枚举的过程也将变得非常慢。正确的方法是使用辗转相除。

The Separator in Grid (PKU2056)

问题简述

有一个 $N \times M$ 的格子，有WSB三种状态。并且S是W和B的分割线。所有的S都可以往右移一格。需要找一条路径，从第一行某个S开始，经过某些S，到达最后行某个S，中间经过的S数最少。

$$3 \leq n, m \leq 200$$

The Separator in Grid

问题分析

这题就是简单的最短路。先预处理，将S和右边那格标记为可走。然后枚举起始点，**BFS**或**DFS**或**dijkstra**都行。

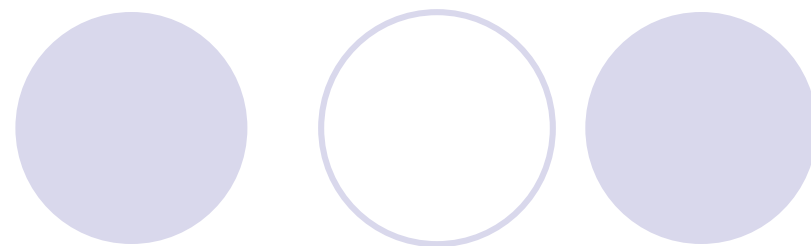
The Lost House (PKU2057)

问题简述

一个N个节点树上，树最多是八叉，有一个蜗牛把壳丢在了某个叶子节点。它从根节点出发沿树枝走去找壳。某些内点上有蚜虫，会告诉你下面叶子上是否有壳。求蜗牛找到壳的路径长度的期望值，即所有可能的路径长度和的平均值。

$$1 \leq N \leq 1000$$

The Lost House

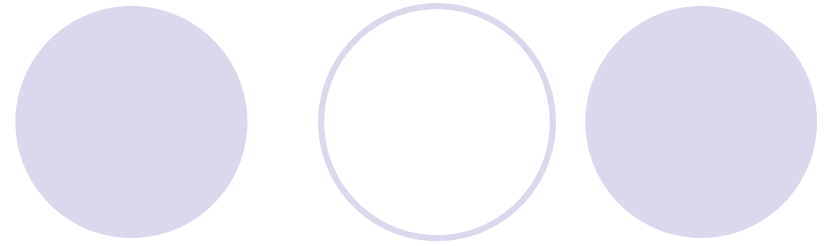


问题分析

这是一道树形**DP**。每个点可以记录三个值：这个点子树中叶节点的个数 $\text{leaf}[i]$ ，如果壳在其子树上蜗牛从该点出发找到壳的数学期望 $\text{fa}[i]$ 和壳不在其子树上蜗牛遍历子树需要的时间 $\text{fb}[i]$ 。

而每个点的 $\text{fb}[i], \text{leaf}[i]$ 很容易通过它的所有子女推得。那么如何推出 $\text{fa}[i]$ 呢？

The Lost House



如果我们已经知道了蜗牛的路线，也就是访问该点子树的顺序 t_1, t_2, \dots, t_k ，那么有

$$fa[i] = \frac{fa[t_1] * leaf[t_1] + (fb[t_1] + fa[t_2]) * leaf[t_2] + (fb[t_1] + fb[t_2] + fa[t_3]) * leaf[t_3] + \dots + (\sum_{i=1}^{k-1} fb[t_i] + fa[t_k]) * leaf[t_k]}{leaf[i]}$$

那么如何确定顺序？由于树枝只有8叉，那么只要利用状态压缩的动态规划， $2^k * k$ 就可以了。