Summer 2013
Contest 6 by Group B-group

Statistics

Problem	AC	WA	PE	RTE	FPE	SF	TLE	MLE	OLE	CE	Submit
A	16	34	0	1	0	1	0	0	1	3	56
B	7	10	0	0	0	0	2	0	0	2	21
C	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	3	1	0	0	4
M	13	21	0	2	0	1	7	2	0	0	46
S	12	55	0	0	1	0	1	0	0	0	69
Summary	48	120	0	3	1	2	13	3	1	5	196

An Involuntary Movement

题目大意：给出一个N个顶点的无向图， 将顶点分成X组， 每组Y个点， 且每一组内的顶点两两互不连通， 求最大分组的数目

解法：

1 二分：因为具体分组只和图的连通分量个数及每个连通分量的顶点数有关， 所以先预处理出各连通分量的情况，然后二分X，检验是否存在分组方案满足X和Y；

2 贪心： 每次取剩余点数最多的Y个分量

企鹅之战

Q宠大乐斗

题目描述

- 有n只企鹅，每一只企鹅都住在一个岛上，不同的企鹅住不同的岛，题目保证任意两个岛之间有且仅有一条路径
- 每只企鹅有一个战斗值（可能为负数）
- 岛屿直接有边相连的企鹅们可以结盟，结盟具有传递性，a和b结盟，b和c结盟，那么abc都是盟友（当然可以有更多的企鹅结盟，或者一只企鹅组成联盟也是可以的）

样例说明

- 3
- Flandre -10
- edward_mj 12
- oldjunyi 11
- Flandre edward_mj
- edward_mj oldjunyi
- 答案为11，只要Flandre和edward_mj结盟，oldjunyi单独组成联盟

4

Flandre 10

edward_mj 60

oldjunyi 60

gaoxuezhang -100

Flandre edward_mj

edward_mj oldjunyi

oldjunyi gaoxuezhang

答案为20，Flandre单独组成联盟，剩下edward_mj,oldjunyi和gaoxuezhang组成联盟

解题思路

- 题目就是给定一棵树，求一种power最小的划分
- 比较显然的是，答案不会超过单只企鹅战斗力的最大值。
- 并且如果ans可行，那么ans+1一定可行；如果ans不可行，ans-1一定不可行
- 可以采用二分答案的做法

Cut the Cake — Description

- 有一块半径为R的蛋糕, 蛋糕的不同位置甜度不一样, 点(x,y)处的甜度可以用函数ρ(x, y)=Ax^2+By^2来表示. 现在要切蛋糕，有两类刀.
- 第一类为圆形(半径为1,2,...,R-1), 切下去之后会把圆形蛋糕分成一个圆环形蛋糕和另一个圆形蛋糕, 会把一个圆环形分成两个圆环形蛋糕, 最多切R-1个位置.
- 第二类为线形(长度恰好为R), 每次都是沿时钟整点方向, 最多切12个位置.
- 我们需要切K次, 每块蛋糕的甜度可以用二重积分搞出来. 然后求一种切法, 使得得到的N块蛋糕的均方差最小.

Cut the Cake — Solution

- 首先, 我们对应的S(i)

$$
\begin{aligned}
S_i &= \int_\alpha^\beta \int_{r_1}^{r_2} \rho(r\cos\theta, r\sin\theta) r\,dr\,d\theta \\
&= \int_\alpha^\beta \int_{r_1}^{r_2} A(r\cos\theta)^2 + B(r\sin\theta)^2 r\,dr\,d\theta \\
&= \int_\alpha^\beta A(\cos\theta)^2 + B(\sin\theta)^2 d\theta \int_{r_1}^{r_2} r^3 dr \\
&= \left(\frac{A+B}{2}\theta + \frac{A-B}{4}\sin 2\theta\Big|_\alpha^\beta\right)\left(\frac{r^4}{4}\Big|_{r_1}^{r_2}\right) \\
&= \frac{r_2^4 - r_1^4}{4}\left(\frac{(A+B)(\beta-\alpha)}{2} + \frac{A-B}{2}\sin(\beta-\alpha)\cos(\alpha+\beta)\right)
\end{aligned}
$$

- 这个式子有一个性质, 可以优化算法, 下面再讲

Cut the Cake — Solution

- 然后, 我们来化简一下这个均方差公式, 得到

$$\sigma^2 = \frac{1}{N}(\sum_{i=1}^{N} S_i^2) - (\overline{S})^2$$

- 注意到, 如果我们能确定N, 答案就只和平方和的大小有关了.
- 关键是我们如何才能确定N?

Cut the Cake — Solution

- 注意观察题目的数据范围, 我们知道第一类刀最多切99次, 而第二类刀最多切12次. 然后就可以想到枚举第二类刀切的位置.

- 假设第二类刀切了a次, 那么N=a*(K-a+1), 这里还需要注意a=0和a=1的情况是一样的, 特判一下就好了.

Cut the Cake — Solution

- 接下来的问题就是如何求最小平方和了.
- 对于这个, 我们很容易想到用dp.
- 设dp[i][r1]表示半径为r1的蛋糕划分为i部分的最小平方和, dp方程很简单：

dp[i][r1]=min{dp[i][r1], dp[i-1][r2]+sum[r2][r1]}

- sum[r2][r1]表示这个圆环用第二类刀划分的平方和, 这个可以在枚举的同时处理出来.
- 这个算法复杂度是O(2^12*R^3), 显然超时

Cut the Cake — Solution

- 为了优化dp, 我们回到那个S(i)的公式

$$
\begin{aligned}
S_i &= \int_\alpha^\beta \int_{r_1}^{r_2} \rho(r\cos\theta, r\sin\theta) r\,dr\,d\theta \\
&= \int_\alpha^\beta \int_{r_1}^{r_2} A(r\cos\theta)^2 + B(r\sin\theta)^2 r\,dr\,d\theta \\
&= \int_\alpha^\beta A(\cos\theta)^2 + B(\sin\theta)^2 d\theta \int_{r_1}^{r_2} r^3 dr \\
&= \left(\frac{A+B}{2}\theta + \frac{A-B}{4}\sin 2\theta \Big|_\alpha^\beta\right)\left(\frac{r^4}{4}\Big|_{r_1}^{r_2}\right) \\
&= \frac{r_2^4 - r_1^4}{4}\left(\frac{(A+B)(\beta-\alpha)}{2} + \frac{A-B}{2}\sin(\beta-\alpha)\cos(\alpha+\beta)\right)
\end{aligned}
$$

- 观察到，括号内部分和r1或者r2没有任何关系. 换句话说就是第一类刀的切法和第二类刀的切法是独立的。

Cut the Cake — Solution

- 然后, 我们就可以在一开始dp预处理一下第一类刀的切法. dp[i][j]表示半径为j, 切i次

  dp[i][j]=min{dp[i][j], dp[i-1][k]+(j^4-k^4)^2}

- 接下来枚举第二类刀的切法，求出式子括号内的部分, 计为ret. 答案就是

  min(sqrt(ret*dp[K-a][R]/N-ave*ave))

- a就是前文讲的那个, ave就是S(i)的平均值
- 这样时间复杂度就是O(100^3+T*2^12*12), T为测试数据组数，能够很快出解.

Cut the Cake — Note

1)由于精度问题, 计算过程会出现根号内为负数的情况, 需要在输出答案的时候特判一下.

2)还是精度问题吧, 在计算过程中会出现根号内是一个差不多1e-12的数, 这个时候答案应该是为0, 但是输出结果和0差了很多, 也需要特判一下.

3)计算二重积分的时候，需要注意θ的范围是[0, 2π], 不过由于计算方法的不同，可能不需要注意这点.

Eibonacci Sequence

$$f(n) = \begin{cases} 0, & n < 0 \\ 1, & n = 0 \\ Af(n-a) + Bf(n-b), & n > 0 \end{cases}$$

Eibonacci Sequence

- 当a,b比较小的时候，可以把线性递推关系表达为矩阵，使用矩阵乘法可以在O(max{a,b}^3*log(N))算出结果。
- 当a,b比较大的时候，则考虑使用生成函数来计算。

$$F(z) = \frac{1}{1 - Az^a - Bz^b} = \sum_{k=0}^{\infty} (Az^a + Bz^b)^k$$

- 计算f(n),就是计算z^n的系数，O(n/max{a,b})

$$\sum_{ap+bq=n} C_{p+q}^{q} A^p B^q$$

Eibonacci Sequence

- 于是对于给定的一组数据可以在两种里面选复杂度低的进行计算，设置一个阈值分别计算。
- 具体计算还可能要用到模运算求逆，中国剩余定理，扩展欧几里德算法等，比较麻烦，阶乘模计算等，但大方向的思路还是比较简单的。

扫雷

题目描述

- 游戏的玩法和普通的扫雷一样
- 给出的矩阵大小是n行（m*2+1）列
- 一开始给出所有的偶数列的信息
- 问满足当前的信息情况下，总共最少的雷的数量

解题思路

- 方法一：用dp[i][mask]表示第i个奇数列，雷的状态为mask的情况下的最少雷的数量，根据行的状态来转移mask
- 时间复杂度O(m*2^(n+n))
- 方法二：枚举整个矩阵的第一行和第一列，由此可以推出整个矩阵的状态
- 时间复杂度O(2^(m+n))

S Count the Sets

By zuxu

Definition

- *T* is a subset of the power set of *A*.
  - |T| is at most 2^n, given |A| = n.
- *T* contains both the empty set and *A*.
  - |T| is at least 2.
- *T* is closed under union and intersection.
  - A = {1, 2, 3, 4}
  - Union only: { E, {1, 2, 3}, {2, 3, 4}, A}
  - Intersection only: { E, {1}, {1, 2}, {1,3}, A}

Details

- m = 1, just impossible…
- m = 2, only one: { E, A }
- m = 3, format: { E, X, A }
  - X can be any subset of A, but ~~E~~ and ~~A~~
  - Thus 2^n - 2
- m = 4, format: { E, X, Y, A }
  - $X \cup Y = A$ && $X \cap Y = E$
  - $E \subseteq X \subseteq Y \subseteq A$

Derivation

- X ∪ Y = A && X∩Y = E
  - (2^n – 2)/2 = 2^(n-1) – 1
- E ⊆ X ⊆ Y ⊆ E

$$
\begin{aligned}
(n) &= (C_n^1 C_{n-1}^1 + C_n^1 C_{n-1}^2 + \cdots + C_n^1 C_{n-1}^{n-2}) + (C_n^2 C_{n-2}^1 + C_n^2 C_{n-2}^2 + \cdots + C_n^2 C_{n-2}^{n-3}) + \cdots + (C_n^{n-3} C_3^1 + \\
&\quad C_n^{n-3} C_3^2) + C_n^{n-2} C_2^1 \\
&= C_n^1 (2^{n-1} - 2) + C_n^2 (2^{n-2} - 2) + \cdots + C_n^{n-3} (2^3 - 2) + C_n^{n-2} (2^2 - 2) \\
&= (C_n^1 2^{n-1} + C_n^2 2^{n-2} + \cdots + C_n^{n-2} 2^2) - 2 (C_n^1 + C_n^2 + \cdots + C_n^{n-2}) \\
&= (\sum_{k=1}^{n-2} C_n^k 2^{n-k}) - 2 (2^n - 2 - n) \\
&= (\sum_{k=1}^{n-2} C_n^k 2^{n-k} + C_n^{n-1} 2^{n-(n-1)} + C_n^0 2^{n-0} + C_n^n 2^{n-n}) - 2 (2^n - 2) - 2^n - 1 \\
&= \sum_{k=0}^{n} C_n^k 2^{n-k} - 3 (2^n - 1) \\
&= (1+2)^n - 3 (2^n - 1) \\
&= 3^n - 3 (2^n - 1)
\end{aligned}
$$

Reference

- “SP-set”: the definition of topology on finite sets in general topology.

- For more results:
- http://en.wikipedia.org/wiki/General_topology
- http://www.emis.ams.org/journals/JIS/VOL9/Benoumhani/benoumhani11.pdf