

## 1. Wordl Finals 2010A APL Lives!

这道题目主要考察的是模拟。要注意几点，

一是“+，-，\*”号的运算规则是类似的，也就是说三种操作符都要分别考虑两种情况，左右操作数是相同形态的数组，或者是两个操作数中存在至少一个是单个元素的一维数组，这两种情况分别有不同的结果。（特别地，因为减号不满足交换律，还要考虑单个元素是左操作数，还是右操作数）；

二是要设计科学的存储方法，这题涉及的计算都是和数组有关的。而且一维、二维、三维数组都有出现，因此我们如果使用多维数组来存储的话不好统一情况，所以应该全部都使用一维数组存储，同时记下每一维的大小。这样二维和三维的数组都可以转换成一维进行操作，可以使用 vector 来方便地动态分配内存。同时各种操作都可以简化。对于“rho”只要直接把右操作数的一维数组存到新开的大小为数组三个维度乘积的一维数组中，不够的位循环补齐。对于“+ /,- / \* /”，假设数组最后一维的大小为  $p$ ，那么就相当于一维数组每连续  $p$  个数都在相邻两数间添加操作符，计算后得到一个数，将得到的这个数按原顺序存到新的一维数组，同时新的数组的维度为原数组的去掉了最后一维。对于“+，-，\*”，可以直接当作一维数组进行计算，新数组的维度继承原数组。输出时按照三种情况进行一些特殊判断即可；

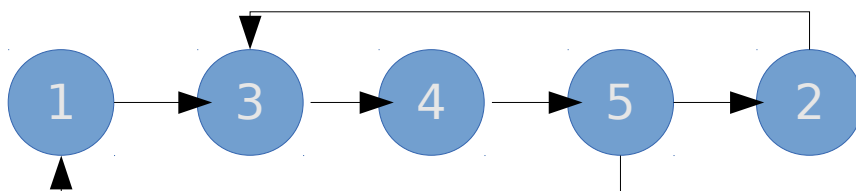
三是要注意表达式计算的方法，这题计算顺序是从右到左，无优先级，这与常规的表达式计算不同，要区别。将读入的字符串按照空格划分后，得到了一个由数字、变量、操作符构成的列表，我们从后往前，首先获取最后一个右操作数，如果遇到“)”就递归处理括号中的表达式并得到一个数组，如果遇到数字就把连续的数字当作一维数组读入，如果是变量就从变量表中查值（可用 map）。这样就得到了右操作数，之后读取操作符，如果是双目操作符还要读左操作数，之后计算，并将答案当作右操作数与下一个操作符进行计算，直到计算完毕。

大致就是注意以上三点就可以 AC.此外这题出数据难度也很大，因为要保证语句合法，而且所有中间值都不超出范围。就先用 rho,drop,iota 构造了 50 个形态一样三维数组，之后用这 50 个变量加上单个数字随机生成一些由+,-,=,\*,括号构成的算式。之后我使用 / 操作符 50 个变量将三维降成二维，再重复之前随机算式的操作。最后再将成一维。这样的好处是因为只要把原来的数字设得很小，同时减少=,\*的数量，就可以保证不越界。但是没有用到 + /,\* /，所以加上了若干专门测试这两个算符的式子。

## 2. Wordl Finals 2006J Routing

这道题目的题意是给定一个有向图，要求出从节点 1 到节点 2 和从节点 2 到节点 1 的路径，使得包含在这两条路径中的节点数目最少（在两条路径中都出现只计算一次），求这个最小值。

我们用  $dp[x][y]$  来表示求出 1 到  $x$  和  $y$  到 1 的路径，包含节点数的最小值。那么结果就是  $dp[2][2]$  了。初始是设置  $dp[1][1] = 1$ ，然后使用 spfa 来进行 dp。假设当前状态为  $x, y$ ，那么枚举下一节点  $z$ ，分别用  $dp[x][y] + 1$  更新  $dp[x][z]$  和  $dp[z][y]$ （如果相关边存在），并注意如果  $x=z$  或者  $y=z$ ，则根据题意中不需加一。但这样存在一个问题，就是当且仅当 dp 过程中两条路径中的公共点恰好都按照顺序出现时才能正确计算，否则会使得答案变大。比如下图中的情况，答案应该是 5。但是因为公共点 3,5 在一条路径中是按照 3,5 的次序出现，另一条路径中按照 5,3 的次序出现，因而 dp 的时候比如会导致有一个点没被当作公共点使得答案变成 6。



因此加上一个特殊情况，即使用  $dp[x][y] + dist[x][y] - 1$  更新  $dp[y][x]$ 。（ $dist[x][y]$  为从  $x$  到  $y$  的最短距离）。即考虑当前路径为 1 到  $x$ ， $y$  到 1，现在给两段路径都添加上  $x$  到  $y$  的这一段（对于  $y$  到 1 的路径相当于反着添加），得到 1 到  $y$ ， $x$  到 1 的两条路径。这样就可以解决了。

## 3. Wordl Finals 2000D Gifts Large and Small

这道题目的题意是给定一个简单多边形，要求出一个紧紧包住多边形的矩形的最大和最小面积是多少。

首先我们如果知道矩形四条边中任意一组对边的倾斜角，那么矩形的面积可以很快求出。方法如下，我们设倾斜角为  $a$ ，那么假设将整个矩形顺时针旋转  $a$ ，就可以使得矩形的边都平行于坐标轴，这样求出旋转后的每个点的坐标。则因为矩形紧紧包住多边形，所以矩形四条边一定分别通过横坐标最大最小，纵坐标最大最小的点。用横坐标的最大值减去最小值得到一条边长，再纵坐标的最大值减去最小值得到一条边长，就可以在  $O(n)$  时间内计算出面积。

现在我们发现矩形的面积是一个只与倾斜角相关的函数，而且不难发现这个函数具有一定的连续性，因此我们可以考虑使用爬山法或者模拟退火的方法来解决。事实上，通过试验可以发现很容易就可以找到最优取值。