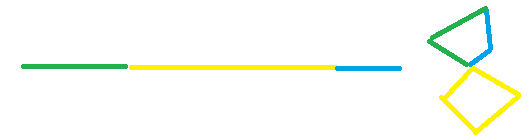
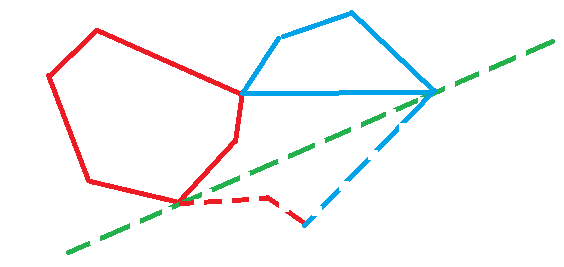
ACM/ICPC Worldfinals 2011 G magicsticks解题报告

南京师大附中 余岳

本题通过分析可以得到以下结论

结论1：最后形成的多边形肯定都是由折线上连续的一段围成的，即不会出现下图的情况：折线上不相邻的两段通过一定的折叠来共同构成一个多边形。

若出现这种情况，那么会出现两个多边形共用同一个顶点，那么我们可以作出多边形的公切线将它们翻折出去，可以得到更大的面积：

结论2：一个顺次给出每条边边长的多边形，在它的所有顶点共圆时面积最大。

首先，在数学上利用余弦定理以及适当的代数变换可以证明四边形的情况。之后，对于多边形A1A2..AN上的边AiAi+1、Ai+1Ai+2、Ai+2Ai+3，可以固定Ai点和Ai+3点，即固定了AiAi+3的长度，那么AiAi+1Ai+2Ai+3构成了一个四边形，所以当多边形面积最大时Ai、Ai+1、Ai+2、Ai+3共圆，即所有点共圆。

利用这两个结论，可以写出一个O(n^3logn)的区间DP，这需要预处理每一个折线区间所能围出最大面积。由于共圆时面积最大，因此我们要计算共圆时的半径r。对于某个折线区间a1a2..am，需要分两种情况：

1.共圆时圆心在多边形内

需要解方程，这个方程的左边关于r是单调的，可以直接使用二分法

2.共圆时圆心在多边形外

设A=max(a1..am)

则需要解方程，不幸的是这个方程左边的函数不单调，但是通过画了很多次图像可以发现这个函数与x正半轴只有一个交点且是单峰的，大致形状如下

因此可以先三分出峰值，然后在单调区间上二分零点，从而求解出r。

但由于n=500且涉及大量反三角函数运算，区间DP根本无法利用卡常数通过大数据。

结论3：计算折线上某一个区间构成的最大多边形面积和时，如果不是用所有的线段直接构成**一个**多边形，那么这段折线上最长的一个线段必然不会出现在最优解的任何一个多边形中。

对于这个结论我不会严格证明，但是可以感性地解释一下：假设我们手上有一个多边形，我给它添一个线段，如果线段相对比较小，那么这个多边形所能构成的面积显然会变大，事实上，只有当添上的这个线段足够大以至于将整个多边形“撑”得很平的时候才会让面积变小。而这种特别长的线段在其他线段之和变小的时候所带来的负面影响将更加明显，如果这条长线短使所有线段一起构成的多边形比较扁的话，让它之和其中的一部分在一起，这个多边形就更扁了。因此这条最长边是不会再被用到的。

利用结论3，我们实际上只需要计算O(n)个线段区间所围城的面积，因此复杂度为O(n^2logn)，漂亮地解决了这个问题。

ACM/ICPC Worldfinals 2006 E Bit Compressor解题报告

本题的思路是动态规划。一个小trick是，在状态设计时一不小心可能会加上“当前1的个数”这一维，状态数就过多了，实际上由于原串中0的个数不会多于压缩后的0的个数，将其改为“当前0的个数”，这配合上“当前长度”后实际上也就表达出了“当前1的个数”，但状态数小了很多。

具体地，f[i,j,k]表示将压缩串的前i位解压后产生了j位数，其中有k个0，满足此条件时的方案数。状态转移时可以使用主动转移，那么需要考虑一下四种转移方式：

1.原串接着是(0)0

2.原串接着是(01)0

3.原串接着是(011)0

4.原串接着是(011..1)0 (多于2个)

前三种转移里压缩串和原串相同，第四种压缩串接着是(01\*\*\*)0。其中1\*\*\*为1个数的二进制表示。

要注意的是一个特殊情况，压缩串接着是0110的话，原串有两种可能：0110和01110。这样复杂度为O(160000\*40\*40)，略微有点多。其实有效状态并没有我们列出的那么多，由于使用主动转移，我们只需要对不为0的f[I,j,k]来进行转移，因此用哈希表来维护这个数组即可降低些许常数在时限内通过。

ACM/ICPC Worldfinals 2001 H Professor Monotonic’s Network解题报告

本题是一道经典题，关于比较网络的问题可以参见《算法导论》，对于比较网络是否是排序网络存在这样一个结论：当取遍00…0（n个0）到11..1（n个1）这2^n种输入，输出都是单调增的，那么对于任何输入，输出也是单调增的。利用这个结论可以直接解决本题。