GCD

期望60分解法：

枚举i,用辗转相除法求出GCD(i,m),然后取最大值

时间复杂度O(n\*log(max(n,m)))

期望100分解法

枚举m的每个因子d,答案就是满足1<=d<=n的d的最大值

时间复杂度O(sqrt(m))

最短路

期望20分解法：

暴力枚举每一条边，每次重新建图，用dijkstra算法算出从1号点到2号点的最短距离，观察最短距离和原图的关系

时间复杂度O(m\*(m+n)log(n)) 期望得分20分

期望40分解法：

先用dijkstra算法在原图中跑出1号点到i号节点的最短距离dist\_1(i)，将所有边反向后用dijkstra算法求出i号点到2号点的最短距离dist\_2(i)；

再沿着最短路径找到从1号点到i号点的方案数f(i)，以及i号点到2号点的方案数g(i)；

如果一条起点为ai，终点为bi，长度为ci的边满足f(ai)\*g(bi)==f(2)且dist\_1(i)+ci+dist\_2(i)==dist\_1(2)则该边是原图中从1号点到2号的最短路径的必经之路。

将第i条边反向后对该边进行分类讨论

1.dist\_2(ai)+dist\_1(bi)+ci < dist\_1(2) 则最短路径的长度变短了

2.dist\_2(ai)+dist\_1(bi)+ci == dist\_1(2) ,则最短路径的长度并没有发生变化

3.dist\_2(ai)+dist\_1(bi)+ci > dist\_1(2)

a.如果该边不是原图最短路径的必经之路，则最短路径的长度并没有发生变化

b.如果该边是原图最短路径的必经之路，则最短路径的长度变长了或者新图中不存在从1号点到2号点的路径

(当不存在从1号点到i号点的路径时，dist\_1(i)为极大值,dist\_2(i)同理)

时间复杂度O((m+n)log(n))，但当路径数量过多时，f值和g值会超出int的储存范围

期望100分解法：

在上述求必经之路的时候用拓扑序或者其他方法求在最短路径构成的图上的桥

时间复杂度O((m+n)log(n))

数列

期望40分解法

预处理：离散化，然后让连续一段值相同的元素合并为一个元素。

正式DP：

显然有个最差策略为每个元素处都切一次，则切的次数为元素的个数-1

相对地来说就是假设全部元素之间就已经切开，要尽量多地合并元素

DP的第一维用来确认当前是合并了值为多少的两个数值段，DP的第二维来记住最后一次合并是合并了哪个位置的两个线段

即DP[i][j] = 对于值为1到i+1的数值段， 最后一次合并为合并a[j]和a[j+1]这两个元素，最多能合并的总次数

而相对应的转移方程就是：

DP[i][j] =max( DP[i-1][j']+1) (合并 a[j'] , a[j'+1] 不会与 合并a[j],a[j+1]冲突)

冲突是指合并 a[i],a[i+1] 的同时也合并a[j],a[j+1] 会导致无法拼接成单调不降的序列，其充要条件是i+1=j且值为a[i+1]的数值段在原序列中出现了不止1次,

空间和时间复杂度都是O(n^2),期望得分40分

期望100分解法

优化：

1.滚动数组优化空间为O(n)

2.因为对于每个i=x，转移的时候只用考虑最大值和次大值，如果最大值和当前状态冲突，则用次大值更新

空间和时间复杂度都是O(n),期望得分100分