休闲题解

**G**

找到图的一个生成树，考虑剩下的边，如果构成了一个二分图，那么只要把生成树和这个二分图黑白染色，这样搞出四种颜色就行了。否则剩下的边中存在一个奇环，删去之后由于生成树的存在仍然联通，输出即可。

**S**

Subtask 1：问 {1,0,0...0} {0,1,0...0} 这样的n个问题即可。

Subtask 2：问 {1,100,10000,10^7,....} 这样的一个问题就够了。

Subtask 3：由于b不够大，把25个数分成3个一组，每组问一次{1,100,10000,0...}即可。

Subtask 4：

考虑把16个数分成2个一组，每组问一次{1,10^49,0...}，这样前一个数的后49位就是对的了，但是前8位可能会被后一个数加在一起，但是如果我们知道了后一个数的最后8位，我们只要把积中扣掉这8位的贡献，就可以知道前一个数了。

那么如果我们问一次{0,1,0,10^8,0,10^16...}这样的问题，就可以知道第一组后一个数的最后8位，从而还原出第一组的数，然后我们从积中扣掉第一组后一个数，就可以知道第二组后一个数的最后8位了，依次类推。

事实上{0,1,0,10^8,0,10^16...}这个问题位数超过了50位，拆成两个询问就行。

**C**

最优方案显然是走到某个点，然后往回走，路上遇到需要吃的餐馆就进去吃。

考虑一个这样的dp：f[i][j]表示不吃1~i-1这些餐馆，一共吃了j克食物的最小代价，则f[i][j]=min(f[i+1][j-a[i]]+a[i]\*i,f[i+1][j])，f[i][a[i]]还可以对i+i+a[i]\*i取min（走到i吃完就回头）。这显然可以滚动数组。

认真观察这个dp，可以注意到我们每次只要更新满足i\*j<=m的j，这是因为我们当前只考虑了i...n这个后缀，每个元素的代价是a[k]\*k>=a[k]\*i，所以如果吃了j克且i\*j>m，代价必然超过m，那么只需更新所有j<=m/i即可。

复杂度O(mlogm+n)，事实上m=1e7都随便跑。