思路题选讲

bzoj3416 Poi2013 Take-out

题意：小F喜欢玩一个消除游戏——take-out。这是一个单人游戏，游戏者的目标是消除初始时给定的一列砖块，从左往右标号为1到n，若两个砖块标号相差1，则它们相邻。每一块砖块要么是黑的，要么是白的，这列砖块里面白砖块的数量是黑砖块的数量的k倍，游戏者可以通过执行移除操作来消除砖块，一步移除操作会将k个白砖块和1个黑砖块从序列中移除，而这些被移除的砖块原来所在的位置用透明砖块所代替，其它砖块的位置不变，一个移除操作是合法的当且仅当：
在这次移除中，任意两个被移除的砖块之间，没有透明砖块，且恰好移走k个白砖块和1个黑砖块 显然一个砖块不能被移除两次 小F的智商不够……他对着面前密密麻麻的砖块看傻了眼…… 你能帮他玩通关么?
Input 第一行两个数 n,k，意义同题目描述 接下来一行一个由’b’和’c’组成的字符串，长度为n，描述这列砖块 第i个砖块如果是黑色的，那么第i个字符为’c’ 否则是白色的，第i个字符为’b’ (注:波兰文中b是bialy的首字母,c是czarny的首字母)
Output 输出n/(k+1)行，每行k+1个数，用空格分开，要求递增 第i行表示第i次移除操作移除砖块的位置集合
保证k+1|n，保证输入数据有解

题解：
这个“中间不能夹着之前消除过的位置”的要求可能有些奇怪，但如果我们倒着想，那就可以把问题改成“消除后把方块拿走，两侧的拼起来成为新的相邻的方块”，这就好思考多了。
接下来如果意识到很重要的一点，就是只要有解，任意时刻肯定有一个连续区间是可以消除的，那么剩下来的就是如何快速构造一组方案的问题了。
（证明：如果每个黑色位置的两侧直接相邻的白色位置个数之和都小于k，则总共的白色位置个数绝对小于黑色位置个数*k，不满足白:黑=k:1的必要条件）
链表法：用链表维护黑色位置和所有的位置，以及每个黑色位置两侧的白色位置各有多少个。每次找到一个两侧白色位置个数之和够k的黑色位置，将其消除，向它的两侧递归查找是否多出了新的可以消除的黑色位置。这样的过程进行到结束，就会把所有的位置消完。
栈法：从左到右扫描入栈，如果遇到末尾k+1个位置可以消除，就立刻消除。（如何判断可以消除？可以用前缀和！）

bzoj1111 [POI2007]**四进制的天平**Wag

题意：现在有一个重量为n（n<=10^1000）的物品，以及数量无限的砝码，砝码的重量都是4的幂，物品放在一架天平的左边，砝码可以放在左边或右边，问最少用多少个砝码可以称出物品的重量，以及有多少种最优的方案（最优就定义为用的砝码个数最少）。

题解：
可以看做想办法把n分解成尽量少个4的整数次幂的和与差。
在方案最优的情况下，一种重量的砝码最多用3个。证明：如果用了>=4个，可以把4个合成更大的一个砝码。
（事实上，最多用两个，证明：如果用了3个，可以把这三个变成一个更大的减去一个这种小的。）
在以上的发现基础上，我们观察一个最后的方案，如果里面只有加法，那很简单；如果里面有减法，减法会造成借位，比如四进制下，

```
1 . . . -1 .
会变成
0 3 3 3  3 0
再举一个例子：
1 . -1 . -2 . -1 .
会变成
0 3  2 3  1 3  3 0
```

在上例中，我们从把1开始，到后一个-1结束这段区间称为处于“借位状态”。
一个最后的方案，一定是由若干简单的加法，以及这样的一些持续性的“借位状态”区间组成。

进入借位状态的方法是在这一位上多加一个1等着后面来借，从下一个位置开始就进入了借位状态，借位状态内，如果想要达到3，什么也不用做，如果想要得到2，需要减一个1，以此类推。离开借位状态的方法是，这一位上多减一个1，从下一个位置开始就离开了借位状态。
所以我们可以从高位到低位进行DP，设置两个数组，分别表示在状态里和不在状态里的情况下的方案。（方案包括最少砝码数和方案数）

```
f[i - 1][0] <- f[i][0] + b[i]
f[i - 1][0] <- f[i][1] + 4 - b[i]

f[i - 1][1] <- f[i][0] + b[i] + 1
f[i - 1][1] <- f[i][1] + 3 - b[i]
```

bzoj1114 [POI2008]**鲁滨逊逃生**Rob

题意：给出一个n*n（n<=2000）地图，上面有障碍、空地和船三种位置，其中船是连成一片的多边形，中间没有空隙，而且船的形状满足这些条件：

1. 其宽度成“梭形”：有一个最宽的地方，往上和往下都严格变窄。
2. 其是个轴对称图形，有一条横的或竖的对称轴。

每一步，你可以移动船，船的一片都会跟着移动，每次移动可以选择四个方向移动一个单位长度。移动的要求是船的任何部分不能走到障碍上，但是可以走出地图，最后的目的是让整个船都离开地图，问最少的步数。

题解：
根据船的形状的两条性质，我们可以推理出，船一定有一个重心，满足它所在的横轴是船最长的横轴，它所在的纵轴是船最长的纵轴。
那么我们可以从给出的船的形状把船的信息用每层的宽度及其相对位置来表示。
我们把问题分解成两部分：先把船的重心移动到地图的边界，再把船从边界移动到地图外。
第一个问题，我们需要预处理重心在地图的每个位置是否合法。
我们一行一行地算。对于一行，每个障碍物都会让这一行的一段连续区间变得不合法。所以如果我们对每个障碍物对这个区间做标记（打差分标记即可）就可以了。但是障碍物的个数可能很多。注意到如果两个障碍物都在我们枚举的这一行的上方的同一列，则下面的肯定比上面的覆盖的区间长。所以我们可以只要下面的。于是我们通过扫描预处理出每个位置往上第一个障碍，往下第一个障碍，则每行要考虑的就只有O(n)个障碍了。总复杂度O(n^2)。然后一个bfs。
对于第二个问题，可以分成从角落离开和从边上离开两种。从边上离开的步数是确定的。从角落离开的步数取决于船的对应方向的阶梯形的形状，只要枚举每个拐角哪个对应的步数少就行了。
所以我们O(n^2)地解决了这个问题。

bzoj1110 [POI2007]**砝码**Odw

题意：n(100000)个背包(大小10^9)，m(100000)个物品(大小10^9)，任意两个物品的大小之间都存在倍数关系，问用所有n个背包最多装下多少物品。

题解：
物品按大小从小到大排序，则小的一定是大的的约数，且不同大小的物品个数不会超过32。
假设最小的物品大小是a，第二小的物品大小是ka，则所有背包的大小除以a的余数部分永远无法使用，删除之。所有背包的大小除以ka的余数无法装入除了最小的物品以外的物品，所以这些部分完全用来装最小的物品，装干净。如果最后剩下了余数部分的背包容量，则丢弃；如果最后剩下了大小为a的物品，则把它们组合成若干个大小为ka的物品（组合出的个数为大小为a的物品个数除以k上取整）。
因为有组合成新物品的操作，所以每个物品现在代表多少个原来的物品会有不同（也就是背包问题里的物品加权了），所以整个算法需要相应修改，改为每类物品维护一个大头的优先队列，优先填充大个物品，也优先把最大的若干个组合成新的物品。

bzoj3414 Poi2013 Inspector

题意：一个公司有n个员工和m条记录，每条记录形如三元组(t,j,i)表示“在t时刻时，第j个员工观察到除了他自己以外，还有i个员工在上班”，每个员工都会且仅会在这一天的一个特定闭区间时间段内上班，问前多少条记录是互不矛盾的。

题解：
首先，通过二分答案转化成一个判定问题，也即给出若干三元组，问它们是否互相矛盾。
这个判定问题可以用贪心构造方案的方式来解决。
首先，把所有人分成两类：曾经出现过的，和未曾出现过的。每个曾出现过的，都有一个区间是一定在上班的（就是他自己写观察记录的最早时间和最晚时间框成的区间），所以可以看成有两个端点可以在一个移动区间内任意向左/向右移动的一个区间。最后的要求是让所有观察的部位的人数满足要求。
然后，我们设置初始状态为每个区间都扩展到底，然后从左到右扫描位置，维护每个线头的类型，这里分成这些种类：
左延伸：已使用、未使用、已右靠（中间状态）、已截断（中间状态）
中间：锁定
右延伸：未使用、已左靠（中间状态）
在需要减少线头数量的时候，优先把左未使用变成已右靠，其次把右未使用变成已左靠，最后把左已使用变成已截断。
需要增加线头数量的时候，只能把一个未曾出现过的人用上，然后增加一个右未使用。

bzoj3415 Poi2013 Price List

题意：一个n个点m条边的无向连通图，每条边的长度都是a，现在再在每对满足dis(a,b)==2a（也就是a,b之间的最短路径恰好经过两条边）的点(a,b)之间新增一条长度为b的无向边。请设计一个算法在这张新图上求单源最短路。

题解：
我们需要先思考一下新图上的最短路的方案可以是什么样子。这和a,b的大小关系有关。
如果b>=2a，则新边用不到，最短路就是只用原来的边的最短路。
如果b<2a，则新边就可以用到了。考虑原图的一条最短路，我们一定可以把每相邻两个边替换成一个b，路径长度就在减小。如果这条最短路的长度是偶数，比如是2n个a组成，则全部替换成b之后，得到的方案（n个b）肯定是最优的，因为如果存在更优的方案用了x条a，y条b满足xa+yb 如果最短路的长度是奇数，且没有全部使用b的从起点到终点的路径，则替换方案也是最优的。这也可以用相似的思路证明。
但如果最短路的长度是奇数，还有可能存在一条全部使用b的从起点到终点的路径，这条路径是可能成为最优解的（可以想象在b远远小于a的时候）。
所以我们剩下一个任务，就是设计一个特殊的算法专门找到全部使用b的路径。
这个算法就很难想，它基于暴力的优化。首先，我们可以用从起点开始的两层bfs找到起点能一步b走到的点集，如果我们有了这个工具，配合bfs就能找到所有使用b的路径。
但直接使用这个工具目前的复杂度极限是O(NM)，可以构造出例子。（比如一个点连接着n−1个点的菊花图，从每个点都需要遍历m条边）
注意到，如果使用bfs，则先访问到的点一定可以用更短的由长度为b的边构成的路径访问到，所以如果我们能在扩展到一个新点后，删除继续扩展这个点的边，则每条有效扩展的边只会被访问一次，这部分的均摊复杂度就会降至O(M)，剩下的就是形成三角形的边我们无法解决，这些边造成的复杂度我们可以用这个上界估计：
$\sum_{i=1}^{n}\min(\mathrm{degree}^2(i),m)$
$\le\sum_{i=1}^{n}\sqrt{\mathrm{degree}^2(i)\times m}$ 原因是$\min(a,b)\le\sqrt{ab}$
$=m\sqrt{m}$
所以复杂度是O(m sqrt(m))
这个估计实际上是合理的，因为一个sqrt(m)个点的完全图能达到这个上界。
所以总复杂度是O(m sqrt(m))。