Dancing Links

舞蹈链

问题1：

集合s有n个成员，现在有m个子集，每个子集包含一些成员，每个成员都属于集合s。在m个子集中选择一些子集组成子集的集合t，使t中包含的成员可以覆盖集合s，即s中所有成员都属于t中的某个子集。

重复覆盖：集合s中的任意成员x至少属于t中的一个子集，允许同时属于两个及以上的子集。例如集合，在子集、、中选择即可重复覆盖s。

精确覆盖：集合s中的任意成员x属于且只属于t中的一个子集，不能出现x不属于t中的任何子集，或者x同时属于t中两个及以上的子集。例如集合，在子集、、中选择即可精确覆盖s。

给定集合s和m个子集，求出其重复覆盖和精确覆盖。

重复覆盖解法：

遍历集合s中每个成员x，若其尚未被包含在t中，则在m个集合中寻找一个包含x的子集加入t中，重复该步骤即可获得重复覆盖。

精确覆盖解法：

求解精确覆盖的算法称为X算法，将集合s中n个成员看作列，将m个子集看作行，组成一个的矩阵d。若子集（其中）包含某个成员（其中），则；若不包含则。对于集合有个成员，还有个子集、、、、、的情况，如图所示：



在这个矩阵d上进行回溯法（我个人认为回溯法和深度优先递归搜索的本质是一样的）即可得到精确覆盖，过程如下：

1. 从0开始遍历集合s中每个成员，对于成员0，遍历所有子集，找到第一个满足的子集，选择该子集作为精确覆盖中的一个子集，包含的其他成员，其他子集不能再包含该子集中出现的成员，因此删掉其他包含的子集和，将也删掉；



1. 从1开始遍历集合s中剩下的成员，对于成员1，遍历剩余子集，找到第一个满足的子集，选择该子集作为精确覆盖中的一个子集，包含的其他成员，其他子集不能再包含该子集中出现的成员，因此删掉其他包含的子集，将也删掉，这时矩阵d已经被删空，并且所有成员都被包含了，说明已经找到了精确覆盖，即为，算法结束；



考虑一种失败的情况：

1. 第1次选择的子集为，删除包含这些成员的子集、；



1. 第2次选择，删除，这时矩阵d已经是空矩阵了，但还剩没有包含，因此选择失败了。这时需要恢复上一次删除的和，选择的理由是其包含，此时继续寻找下一个包含的子集，无法找到，则继续恢复上一次删除的、、（这时d已经恢复为原始矩阵了），选择的理由是其包含，寻找下一个包含的子集，在的基础上可以进行下一波循环，最终找到正确的精确覆盖，算法结束；

回溯法的递归结束条件是矩阵d为空（当矩阵d为空时递归结束），每次递归时选择矩阵中的一列（其中），遍历矩阵d中的所有行，找到一行（其中）满足，选择该行。由于精确覆盖的要求，其他包含该行中成员的行，都不能再选择，因此将所有包含该行成员的其他行删掉，然后将行删掉。重复这个操作试图将矩阵d删空，并且在矩阵d变为空矩阵的时候恰好所有成员也都被已经选择的子集覆盖到。在选取包含成员的行时，可能有多个选择，若选择其中一个子集而无法将d删空，则在递归函数中返回这一层，重新尝试下一个子集。

十字链表是一种方便删除矩阵d中的行列、以及恢复行列的数据结构。每个节点有上下左右4个指针指向周围的节点。如图所示，删除节点B时只需要重置为，重置为，而不需要删除和的链接：





恢复节点B时只需要将B插入其指针指向的两个节点A和C之间即可。

舞蹈链算法在最坏情况下的时间复杂度与递归的时间复杂度一样，为。