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都是递归）即可得到精确覆盖，过程如下：

1. 从1开始遍历集合s中每个成员，对于成员1，遍历所有子集，找到第一个满足的子集，即时有，选择该子集作为精确覆盖中的一个子集，已经覆盖的成员有，中已经包含的成员其他子集不能再出现，因此删掉其他包含的子集、、、、，将也删掉；



1. 这时矩阵d中所有子集都被删除，成为空矩阵，但并没有完全覆盖集合s中所有成员，因此(1)的选择是失败的，撤销(1)中的所有操作。继续从1开始遍历集合s中每个成员，对于成员1，遍历所有子集，找到第二个满足的子集，即时有，选择该子集作为精确覆盖中的一个子集，已经覆盖的成员有，中已经包含的成员其他子集不能再出现，因此删掉其他包含的子集、、，将也删掉；



1. 从2开始遍历集合s中剩下的成员，对于成员2，遍历剩余子集，找到第一个满足的子集，即时有，选择该子集作为精确覆盖中的一个子集，已经覆盖的成员有，中已经包含的成员其他子集不能再包含，因此删掉其他包含的子集（没有找到），将也删掉；



1. 从5开始遍历集合s中剩下的成员，对于成员5，遍历剩余子集，找到第一个满足的子集，即时有，选择该子集作为精确覆盖中的一个子集，已经覆盖的成员有，中已经包含的成员其他子集不能再包含（没有找到），将删掉后矩阵d即为空矩阵，并且已经完全覆盖了子集s中的所有成员，则精确覆盖的结果为，算法结束；



1. 当算法进行到矩阵d为空矩阵，但集合s中所有成员并没有被完全覆盖的情况时，说明某一次的子集选择有错误，将该次选择的操作进行恢复，并寻找下一个覆盖要求的子集，继续尝试，直到找到精确覆盖；

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中的所有子集（行），找到一子集（其中）满足，选择该子集（行）。由于精确覆盖的要求，其他包含该子集中任意成员的子集，都不能再选择，将其删掉，并将子集也删掉。重复这个操作直到将矩阵d删空，检查矩阵d为空时是否集合s的所有成员都被覆盖到。在选取包含某个成员的子集时，可能有多个选择，若选择其中一个子集无法最终将集合s完全覆盖，则在递归函数中返回这一层，尝试其他子集，直到找出精确覆盖。

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。对于集合有个成员，还有个子集、、、、、的情况，建立十字链表时需要额外对每一列添加头节点，并添加一个总的head节点连接所有列的头节点，如图所示：

对于如图所示，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。