题解：

本题题目大意在一个01方阵中找出四条边全都是1的正方形的个数，对于正方形内部则没有要求。

一个直观的想法是首先用N^2的时间预处理出每一个是1的点向上下左右四个方向能够延伸的1的最大长度，记为四个数组l, r, u, d。然后我们观察到正方形有一个特征是同一对角线上的两个顶点在原方阵的同一条对角线上。于是我们可以想到枚举原来方阵的每条对角线，然后我们对于每条对角线枚举对角线上所有是1的点i，那么我们可以发现可能和i构成正方形的点应该在该对角线的 [i, i + min(r[i], d[i]) – 1] 闭区间内， 而在这个区间内的点 j 只要满足 j – i + 1 <= min(l[j], u[j]) 也就是满足  
j – min(l[j], u[j]) + 1 <= i，这样的 (i, j) 就能构成一个正方形。

也就是说对于每条对角线，我们可以构造一个数组 a, 使得  
a[i] = i – min(l[i], u[i]) + 1

然后对这个数组有若干次查询，每次查询的是区间 [i, i + min(r[i], d[i]) – 1]内有多少个数满足 a[j] <= i，所有这些问题答案的和就是该问题的结果。对于这个问题，我们可以通过离线算法，先保存所有查询的区间端点，并对所有端点排序。然后使用扫描线算法，如果扫描到的是第i次查询的左端点，就让当前结果减去当前扫描过的数中 <= i的个数，如果扫描到的是第i次查询的有短点，则让当前结果加上当前扫描过的数中 <= i的个数，最后所有结果相加即可。维护当前数出现的个数可以使用树状数组。这样对于每条对角线求结果的复杂度为O(nlogn)，算法总的复杂度为O(n^2logn)。