简单题，就是考虑的东西有些繁杂。设dp[t][x][y]代表是否能在时间t到达点(x,y)，如果可以则dp[t][x][y]=true，否则dp[t][x][y]=false。这样我们只要找到一个最小的，使得dp[][m][n]为true即可。在计算过程中，我们需要计算是否有炮弹到达(x,y)，这样时间复杂度为O(d\*m\*n\*k)。经过分析我们发现，对于一点(x,y)最多只有4个碉堡的子弹能够到达，即其上、下、左、右四个方向距离点(x,y)最近的四个碉堡。如果事先进行预处理，即计算出与每个点相关联的最近4个碉堡，则在以后的计算过程中检查子弹的时候时间复杂度变成O(1)。这样预处理O(m\*n\*k)，计算dp复杂度为O(d\*m\*n)。空间上可以用滚动数组进行优化。

在这里，计算某点(x,y)在t时刻是否有子弹威胁有个小技巧。因为xiaoz的速度为1，所以xiaoz到达任何坐标的时间均为整数，而且由于每发炮弹的时间间隔也是整数，设碉堡距离(x,y)的距离为L，若L/ 不是整数，那么该碉堡对(x,y)没有威胁。否则判断当前时间t，若 (t- L/)% = 0，则该碉堡在t时刻对于点(x,y)有威胁，否则无威胁。