Analysis

Let's consider the graph with vertices the accessible cells and edges the possible jumps of the knight. From this point on, I'm going to consider this graph.

One important and easy thing to notice is that the graph is bipartite.

It turns out for this problem that it is easier to consider the case for K=1 because it's easier to wrap our heads around it. For larger values of K it turns out analogous.

Considering the case for *K*=1, the following 2 observations are fairly obvious: each vertex (i.e. each accessible cell) should be shot at least once (otherwise the knight could just stay there) and there is no use in shooting any cell more than twice (1 shot to "force" the knight to jump out of there, if it is there, and one shot to destroy it if it has jumped from somewhere else to this cell).

Hence, the task is to find a sequence of shots such that minimal number of cells are shot twice.

We can notice that each cell, which is hit only once has to have only neighbours which are shot twice (once before and once after). This means that the set of vertices which are shot twice have to cover the whole graph via the edges they take part in. We also want this set to be minimal. This is the famous task *minimal vertex cover*. In the general case this task has no polynomial solution, ***but the graph we are currently considering is bipartite***.

A solution of minimal vertex cover for bipartite graphs follows.

The requirement that we need to cover each edge of the graph, naturally gives us a lower bound for the size of the cover - the size of the maximal matching. Why? Well the maximal matching is a maximal set of disjoint edges (edges with no common vertex). For each of these edges, at least one of the ends of the edge have to be chosen in the cover. *Thus, the number vertices in the cover is at least the size of the maximal matching*.

It turns out, that this number is enough. Why? There are a lot of proofs as to why. One could could do it constructively quite easily, considering how to cover the edges outside the matching. I however will use a different way. I will use the famous fact that the maximal flow equals the minimal cut. If we consider a minimal cut of our graph with attached source and sink to respectively the two parts of the bipartite graph, it will contain sets of edges from the source and edges to the sink. Exactly the other ends of these edges give us a cover. That's because if after removing this vertices there is still an edge left there is still a path from source to sink and thus that's not a cut. Thus the size of the minimal vertex cover equals the size of the maximal matching for bipartite graphs, and which are the vertices in the cover we can deduce from the minimal cut or one of the other ways to prove the statement.

Thus, we can find the set of vertices to shoot twice with a flow for time O(N*M*sqrt(N*M)) using the Dinic algorithm. Let the minimal vertex cover be the set C. Then the solution is to shoot C, Than the vertices from V\C (V is the set of all vertices) and finally to shoot C again.

Everything so far was for *K*=1. What about larger K? First, let's not consider the vertices with no edges. They should be shot only once and we could forget about them. Analogically to K=1 are proved the following two statements, that the vertices should be shot at least (K+1)/2 times and not more than (K+3)/2. (Interesting here is that for K-even only one whole number remains in the interval). Similarly, the ones that are shot (K+3)/2 times (rounded down) have to form a minimal cover. And the resulting sequence is formed by alternatingly shooting at the cover C and V\C.

Author: Ivo Dilov