C8. Alice has a map of Wonderland, a country consisting of $n \geqslant 2$ towns. For every pair of towns, there is a narrow road going from one town to the other. One day, all the roads are declared to be "one way" only. Alice has no information on the direction of the roads, but the King of Hearts has offered to help her. She is allowed to ask him a number of questions. For each question in turn, Alice chooses a pair of towns and the King of Hearts tells her the direction of the road connecting those two towns.

Alice wants to know whether there is at least one town in Wonderland with at most one outgoing road. Prove that she can always find out by asking at most $4n$ questions.

Comment. This problem could be posed with an explicit statement about points being awarded for weaker bounds cn for some $c > 4$, in the style of IMO 2014 Problem 6.

(Thailand)

Solution. We will show Alice needs to ask at most $4n - 7$ questions. Her strategy has the following phases. In what follows, S is the set of towns that Alice, so far, does not know to have more than one outgoing road (so initially $|S| = n$).

Phase 1. Alice chooses any two towns, say A and B. Without loss of generality, suppose that the King of Hearts' answer is that the road goes from A to B.

At the end of this phase, Alice has asked 1 question.

Phase 2. During this phase there is a single (variable) town T that is known to have at least one incoming road but not yet known to have any outgoing roads. Initially, T is B. Alice does the following $n - 2$ times: she picks a town X she has not asked about before, and asks the direction of the road between T and X. If it is from X to T, T is unchanged; if it is from T to X, X becomes the new choice of town T, as the previous T is now known to have an outgoing road.

At the end of this phase, Alice has asked a total of $n - 1$ questions. The final town T is not yet known to have any outgoing roads, while every other town has exactly one outgoing road known. The undirected graph of roads whose directions are known is a tree.

Phase 3. During this phase, Alice asks about the directions of all roads between T and another town she has not previously asked about, stopping if she finds two outgoing roads from T. This phase involves at most $n - 2$ questions. If she does not find two outgoing roads from T, she has answered her original question with at most $2n - 3 \leqslant 4n - 7$ questions, so in what follows we suppose that she does find two outgoing roads, asking a total of k questions in this phase, where $2 \leqslant k \leqslant n - 2$ (and thus $n \geqslant 4$ for what follows).

For every question where the road goes towards T, the town at the other end is removed from S (as it already had one outgoing road known), while the last question resulted in T being removed from S. So at the end of this phase, $|S| = n - k + 1$, while a total of $n + k - 1$ questions have been asked. Furthermore, the undirected graph of roads within S whose directions are known contains no cycles (as T is no longer a member of S, all questions asked in this phase involved T and the graph was a tree before this phase started). Every town in S has exactly one outgoing road known (not necessarily to another town in S).

Phase 4. During this phase, Alice repeatedly picks any pair of towns in S for which she does not know the direction of the road between them. Because every town in S has exactly one outgoing road known, this always results in the removal of one of those two towns from S. Because there are no cycles in the graph of roads of known direction within S, this can continue until there are at most 2 towns left in S.

If it ends with t towns left, $n - k + 1 - t$ questions were asked in this phase, so a total of $2n - t$ questions have been asked.

Phase 5. During this phase, Alice asks about all the roads from the remaining towns in S that she has not previously asked about. She has definitely already asked about any road between those towns (if $t = 2$). She must also have asked in one of the first two phases about

at least one other road involving one of those towns (as those phases resulted in a tree with $n > 2$ vertices). So she asks at most $t(n-t)-1$ questions in this phase.

At the end of this phase, Alice knows whether any town has at most one outgoing road. If $t = 1$, at most $3n - 3 \leqslant 4n - 7$ questions were needed in total, while if $t = 2$, at most $4n - 7$ questions were needed in total.

Comment 1. The version of this problem originally submitted asked only for an upper bound of $5n$, which is much simpler to prove. The Problem Selection Committee preferred a version with an asymptotically optimal constant. In the following comment, we will show that the constant is optimal.

Comment 2. We will show that Alice *cannot* always find out by asking at most $4n - 3(\log_2 n) - 15$ questions, if $n \geqslant 8$.

To show this, we suppose the King of Hearts is choosing the directions as he goes along, only picking the direction of a road when Alice asks about it for the first time. We provide a strategy for the King of Hearts that ensures that, after the given number of questions, the map is still consistent both with the existence of a town with at most one outgoing road, and with the nonexistence of such a town. His strategy has the following phases. When describing how the King of Hearts' answer to a question is determined below, we always assume he is being asked about a road for the first time (otherwise, he just repeats his previous answer for that road). This strategy is described throughout in graph-theoretic terms (vertices and edges rather than towns and roads).

Phase 1. In this phase, we consider the undirected graph formed by edges whose directions are known. The phase terminates when there are exactly 8 connected components whose undirected graphs are trees. The following invariant is maintained: in a component with k vertices whose undirected graph is a tree, every vertex has at most $\lfloor \log_2 k \rfloor$ edges into it.

- If the King of Hearts is asked about an edge between two vertices in the same component, or about an edge between two components at least one of which is not a tree, he chooses any direction for that edge arbitrarily.

- If he is asked about an edge between a vertex in component A that has a vertices and is a tree and a vertex in component B that has b vertices and is a tree, suppose without loss of generality that $a \geqslant b$. He then chooses the edge to go from A to B. In this case, the new number of edges into any vertex is at most $\max\{\lfloor \log_2 a \rfloor, \lfloor \log_2 b \rfloor + 1\} \leqslant \lfloor \log_2(a+b) \rfloor$.

In all cases, the invariant is preserved, and the number of tree components either remains unchanged or goes down by 1. Assuming Alice does not repeat questions, the process must eventually terminate with 8 tree components, and at least $n - 8$ questions having been asked.

Note that each tree component contains at least one vertex with no outgoing edges. Colour one such vertex in each tree component red.

Phase 2. Let V_1, V_2 and V_3 be the three of the red vertices whose components are smallest (so their components together have at most $\left\lfloor \frac{3}{8} n \right\rfloor$ vertices, with each component having at most $\left\lfloor \frac{3}{8} n - 2 \right\rfloor$ vertices). Let sets C_1, C_2, ... be the connected components after removing the V_j. By construction, there are no edges with known direction between C_i and C_j for $i \neq j$, and there are at least five such components.

If at any point during this phase, the King of Hearts is asked about an edge within one of the C_i, he chooses an arbitrary direction. If he is asked about an edge between C_i and C_j for $i \neq j$, he answers so that all edges go from C_i to C_{i+1} and C_{i+2}, with indices taken modulo the number of components, and chooses arbitrarily for other pairs. This ensures that all vertices other than the V_j will have more than one outgoing edge.

For edges involving one of the V_j he answers as follows, so as to remain consistent for as long as possible with both possibilities for whether one of those vertices has at most one outgoing edge. Note that as they were red vertices, they have no outgoing edges at the start of this phase. For edges between two of the V_j, he answers that the edges go from V_1 to V_2, from V_2 to V_3 and from V_3 to V_1. For edges between V_j and some other vertex, he always answers that the edge goes into V_j, except for the last such edge for which he is asked the question for any given V_j, for which he answers that the

edge goes out of V_j. Thus, as long as at least one of the V_j has not had the question answered for all the vertices that are not among the V_j, his answers are still compatible both with all vertices having more than one outgoing edge, and with that V_j having only one outgoing edge.

At the start of this phase, each of the V_j has at most $\left\lfloor \log_2 \left\lfloor \frac{3}{8}n - 2 \right\rfloor \right\rfloor < (\log_2 n) - 1$ incoming edges. Thus, Alice cannot determine whether some vertex has only one outgoing edge within $3(n - 3 - ((\log_2 n) - 1)) - 1$ questions in this phase; that is, $4n - 3(\log_2 n) - 15$ questions total.

Comment 3. We can also improve the upper bound slightly, to $4n - 2(\log_2 n) + 1$. (We do not know where the precise minimum number of questions lies between $4n - 3(\log_2 n) + O(1)$ and $4n - 2(\log_2 n) + O(1)$.) Suppose $n \geqslant 5$ (otherwise no questions are required at all).

To do this, we replace Phases 1 and 2 of the given solution with a different strategy that also results in a spanning tree where one vertex V is not known to have any outgoing edges, and all other vertices have exactly one outgoing edge known, but where there is more control over the numbers of incoming edges. In Phases 3 and 4 we then take more care about the order in which pairs of towns are chosen, to ensure that each of the remaining towns has already had a question asked about at least $\log_2 n + O(1)$ edges.

Define trees T_m with 2^m vertices, exactly one of which (the *root*) has no outgoing edges and the rest of which have exactly one outgoing edge, as follows: T_0 is a single vertex, while T_m is constructed by joining the roots of two copies of T_{m-1} with an edge in either direction. If $n = 2^m$ we can readily ask $n - 1$ questions, resulting in a tree T_m for the edges with known direction: first ask about 2^{m-1} disjoint pairs of vertices, then about 2^{m-2} disjoint pairs of the roots of the resulting T_1 trees, and so on. For the general case, where n is not a power of 2, after k stages of this process we have $\lfloor n/2^k \rfloor$ trees, each of which is like T_k but may have some extra vertices (but, however, a unique root). If there are an even number of trees, then ask about pairs of their roots. If there are an odd number (greater than 1) of trees, when a single T_k is left over, ask about its root together with that of one of the T_{k+1} trees.

Say $m = \lfloor \log_2 n \rfloor$. The result of that process is a single T_m tree, possibly with some extra vertices but still a unique root V. That root has at least m incoming edges, and we may list vertices V_0, $\ldots$, V_{m-1} with edges to V, such that, for all $0 \leqslant i < m$, vertex V_i itself has at least i incoming edges.

Now divide the vertices other than V into two parts: A has all vertices at an odd distance from V and B has all the vertices at an even distance from B. Both A and B are nonempty; A contains the V_i, while B contains a sequence of vertices with at least 0, 1, $\ldots$, $m - 2$ incoming edges respectively, similar to the V_i. There are no edges with known direction within A or within B.

In Phase 3, then ask about edges between V and other vertices: first those in B, in order of increasing number of incoming edges to the other vertex, then those in A, again in order of increasing number of incoming edges, which involves asking at most $n - 1 - m$ questions in this phase. If two outgoing edges are not found from V, at most $2n - 2 - m \leqslant 4n - 2(\log_2 n) + 1$ questions needed to be asked in total, so we suppose that two outgoing edges were found, with k questions asked in this phase, where $2 \leqslant k \leqslant n - 1 - m$. The state of S is as described in the solution above, with the additional property that, since S must still contain all vertices with edges to V, it contains the vertices V_i described above.

In Phase 4, consider the vertices left in B, in increasing order of number of edges incoming to a vertex. If s is the least number of incoming edges to such a vertex, then, for any $s \leqslant t \leqslant m - 2$, there are at least $m - t - 2$ vertices with more than t incoming edges. Repeatedly asking about the pair of vertices left in B with the least numbers of incoming edges results in a single vertex left over (if any were in B at all at the start of this phase) with at least $m - 2$ incoming edges. Doing the same with A (which must be nonempty) leaves a vertex with at least $m - 1$ incoming edges.

Thus if only A is nonempty we ask at most $n - m$ questions in Phase 5, so in total at most $3n - m - 1$ questions, while if both are nonempty we ask at most $2n - 2m + 1$ questions in Phase 5, so in total at most $4n - 2m - 1 < 4n - 2(\log_2 n) + 1$ questions.