A few Adversary Arguments

Theorem At least $\binom{n}{2}$ adjacency questions are necessary (in worst case) to determine whether a graph G on n vertices is connected. (Recall by "adjacency" question we mean a question of the form: "is vertex u adjacent to vertex v".)

Proof: Consider any algorithm for this problem, and start it on an unspecified input graph G with n vertices. The Daemon's strategy is to answer no to any edge probe, unless that answer would prove that G is disconnected. More precisely, the Daemon maintains two edge sets X and Y, where initially Y is empty and X contains all $\binom{n}{2}$ edges in K_n, the complete graph on n vertices. The Daemon then performs the following algorithm when an edge e is probed:

<u>Probe(e)</u>
1. if $X-e$ is connected
2. $\quad\quad X \leftarrow X-e$
3. $\quad\quad$ answer No
4. else
5. $\quad\quad Y \leftarrow Y+e$
6. $\quad\quad$ answer Yes

Here we abuse notation slightly and identify the edge set X with the subgraph of K_n consisting of the edges in X together with all vertices in K_n (and similarly for Y.) Observe that at all times $Y \subseteq X$ and the set $X-Y$ consists of precisely those edges of K_n which have not yet been probed. Furthermore both X and Y are consistent with the Daemon's entire sequence of answers since whenever the answer yes is given, that edge is added to Y and remains in X, while if no is given the corresponding edge is removed from X and is not added to Y. The following invariants are maintained over any sequence of edge probes.

(a) The subgraph X is always connected. This is obvious from the construction.
(b) If X contains a cycle, then none of it's edges belong to Y. **Proof:** Deleting an edge from that cycle would leave X connected, and so that edge could not have been added to Y.
(c) It follows from (b) that Y is acyclic.
(d) If $Y \neq X$ then Y is disconnected. **Proof:** Assume, to get a contradiction, that Y is connected. Then being acyclic Y is a tree. Since $Y \neq X$, there exists an edge $e \in X$ with $e \notin Y$. If e were added to Y it would form a cycle with some of the other edges in Y. (This is a well known and obvious property of trees: joining vertices by a new edge creates a unique cycle.) Since $Y \subseteq X$, that cycle is also contained in X. In other words X contains a cycle consisting of e together with some edges in Y. This contradicts remark (b) above. The only way to avoid this contradiction is to conclude that Y is disconnected.

Now suppose the algorithm halts and returns a verdict (connected/disconnected) after probing fewer than $\binom{n}{2}$ edges. Then at least one edge of K_n was not probed, hence $X-Y \neq \varnothing$, and therefore $Y \neq X$. Now (d) tells us that Y is disconnected, and by (a) X is connected. Since both graphs are consistent with the Daemon's answers, the algorithm cannot be considered correct. If the algorithm says G is connected, then the Daemon can claim $G = Y$, while if the algorithm says G is disconnected, the Daemon may claim that $G = X$. Thus any correct algorithm solving this problem must probe all $\binom{n}{2}$ potential edges. ///

Exercise Show that at least $\binom{n}{2}$ 'adjacency' questions are necessary to determine whether a graph G on n vertices is acyclic.

Exercise Let $b = x_1x_2x_3x_4x_5$ be a bit string of length 5, i.e. $x_i \in \{0,1\}$ for $1 \le i \le 5$. Consider the problem of determining whether b contains three consecutive zeros, i.e. whether or not b contains the substring 111. We restrict our attention to those algorithms whose only allowable operation is to peek at a bit. Obviously 5 peeks are sufficient. A decision tree argument provides the (useless) fact that at least one peek is necessary.

a. Use an adversary argument to show that 4 peeks are necessary in general.
b. Design an algorithm which solves the problem using only 4 peeks in worst case. Express your algorithm as a decision tree.