This problem can be decided in polynomial time. It helps to view the $QSAT$ instance as a $CSAT$ instance instead: Player 1 controls the set A of odd-indexed variables while Player 2 controls the set B of even-indexed variables. Our question then becomes: can Player 1 force a win?

We claim that Player 1 can force a win if and only if each clause C_i contains a variable from A. If this is the case, Player 1 can win by setting all variables in A to 1. If this is not the case, then some clause C_i has no variable from A. Player 2 can then win by setting all variables in B to 0: in particular, this will cause the clause C_i to evaluate to 0.

[1] ex63.946.695

We show that *Competitive 3-SAT* $\leq_P$ *Geography on a Graph* by encoding the Boolean formula Φ over which the players are competing as a directed graph G. The construction we use is depicted in the accompanying figure.

For each variable x_i, we create a "diamond" consisting of vertices labeled $a_i, x_i, \overline{x_i}, b_i$ with directed edges as in the figure. The starting node for the game is a_1, and there is an edge (a_{i+1}, b_i) for each $i < n$. Thus, in the initial moves of the game, Player 1 essentially chooses a side of the i^{th} diamond, for each odd value of i, while Player 2 choose a side of the i^{th} diamond for each even value of i. At the end of this process, the current node is b_n, and it is Player 2's turn to move.

The next two moves of the game will now determine which player wins. Below b_n, we create a node c_i, $i = 1, \ldots, k$, representing the k clauses of Φ. Out of each node c_i, we construct edges to nodes associated with terms occurring in clause i, as follows: if x_j occurs in clause i, then we add the edge (c_i, x_j); and if $\overline{x_j}$ occurs in clause i, then we add the edge $(c_i, \overline{x_j})$. These edges model the following construction: Player 2 chooses a clause of Φ, and Player 1 must then choose a term in this clause that has been set to T by the earlier play. If he can do this for any clause, then he must have had a winning strategy for the *Competitive 3-SAT* instance; otherwise, Player 2 must have had a winning strategy. We make this concrete in the following claim.

(1) *Player 1 has a forced win in the* Competitive 3-SAT *instance if and only if Player 1 has a forced win from a_1 in the* Geography on a Graph *instance on G.*

Proof. First suppose that Player 1 has a forced win in the *Competitive 3-SAT* instance. Then he applies this strategy in the *Geography on a Graph* instance as follows. Each time Player 2 moves to a node x_j (respectively $\overline{x_j}$), Player 1 interprets this as setting $x_j = F$ (respectively $x_j = T$), i.e., the true value is the one that is left blank for later use. When Player 1 has to decide whether to move to node x_i or $\overline{x_i}$, he consults his *Competitive 3-SAT* strategy. If it calls for him to set $x_i = T$, then he moves to $\overline{x_i}$; if it calls for him to set $x_i = F$, then he moves to x_i.

Now consider the final two moves of the game. Player 2 chooses the node c_i. We know, since Player 1's choices forced a win in the *Competitive 3-SAT* instance, that there is a term t in c_i that evaluates to T. This means that in the earlier moves, someone chose the node labeled with the negation of t, but no one chose the node labeled t. Consequently, Player 1 can take the edge (c_i, t), and then Player 2 will not be able to make a move.

To show the converse direction, suppose that Player 1 has a forced win in the *Geography on a Graph* instance. Then we can convert this to a *Competitive 3-SAT* strategy in a very similar way: moves to x_i (respectively $\overline{x_i}$) in *Geography on a Graph* correspond to setting $x_i = F$ (respectively $x_i = T$) in the *Competitive 3-SAT* instance. Now, because Player 1 can force a win in *Geography on a Graph*, he must have a legal move for every choice of a node c_i by Player 2. This means that in every clause of the *Competitive 3-SAT* instance, there is some term that evaluates to T — so Player 1 has won the *Competitive 3-SAT* instance. ■

[1]ex554.747.411

We label the vertices $v_1, v_2, \ldots, v_n$ according to a topological ordering. We now define $Win(j)$ to be equal to 1 if the player whose turn it is to move can force a win starting at node v_j, and define $Win(j)$ to be equal to 0 if the other (who isn't about to move) can force a win starting at node v_j.

We can initialize $Win(j) = 0$ for every node v_j with no out-going edges. In particular, this means that we will set $Win(n) = 0$. We now use dynamic programming to compute the values of $Win(j)$ in descending order of j. When we get to a particular value of j, we may assume that we have already computed $Win(k)$ for all $k > j$. Now, a player starting from v_j can force a win if and only if there is some node v_k for which (v_j, v_k) is an edge and a player starting from v_k has a forced loss. Thus, $Win(j) = 1$ if and only if $Win(k) = 0$ for some k with (v_j, v_k) an edge; and otherwise $Win(j) = 0$.

We thus compute all these values in $O(n)$ time per entry, for a total of $O(n^2)$. We then simply check the value of $Win(j)$ for the node v_j on which the game is designated to start.

[1] ex701.675.797