1. (a) To reduce BoxDepth to Clique, we take an arbitrary instance of BoxDepth and convert it into an instance of Clique. Specifically (1) we construct a graph with one vertex for each rectangle, and (2) for each pair of rectangles that intersect, we add an edge in the graph between the corresponding vertices. Assuming we are given n rectangles in BoxDepth, then we note that (1) and (2) induce a polynomial transformation on the size of the original problem instance—n. This is because the construction of the vertices in (1) takes $\Theta(n)$, and adding an edge between any two vertices in (2) amounts to deciding whether or not their corresponding rectangles intersect. This is a task that can be done computationally in $\mathcal{O}(1)$ via intersecting the intervals projected on x-axis and y-axis. Since, we are given n rectangles initially, then we are guaranteed exactly $n(n-1)/2$ decision problems in (2), each of which incurs a constant cost, yielding a total transformation cost of $\mathcal{O}(n^2)$ for (2). Now, since (1) and (2) are polynomial in n, our transformation is guaranteed to be polynomial as well in the size of the original problem instance—n.

We now show why an instance of BoxDepth and an instance of Clique are equivalent to one another via reduction.

Proof. We include both directions and demonstrate that an instance of BoxDepth has a depth of k *iff* the corresponding vertices of their intersecting rectangles form a clique of size k on the graph induced by (1) and (2).

$\Longrightarrow$ (*Sufficient Condition*) — follows trivially from the definition of (1) and (2). Specifically, if an instance of BoxDepth has a depth of k—meaning that k rectangles intersect in some common point—then clearly the corresponding vertices in the constructed graph will have a clique of size k. This is because if a common point belongs to the pairwise intersection of any two rectangles, then any two corresponding vertices are connected by an edge, and thus the k vertices (representing the k intersecting rectangles) form a clique in the graph induced by (1) and (2).

$\Longleftarrow$ (*Necessary Condition*) — conversely, if the constructed graph induced by (1) and (2) forms a clique of size k, with $k > 1$, then we are guaranteed that the corresponding rectangles are in such way that every pair of them intersect. But since every pair of them intersect then all of the projections of the rectangles on the x-axis and y-axis also intersect pairwise, and we can use **Helly's Theorem**[1] in $\mathbb{R}$ to conclude that the rectangles have a common intersection and hence have a depth of k. For special cases of cliques of size $k = 0$ and $k = 1$, the converse follows vacuously (when $k = 0$) and trivially (when $k = 1$) from the definition of BoxDepth.

□

(b) *Algorithm Description and Time Analysis:*

We consider all the projections of the rectangles on the x-axis and y-axis. There are at most $2n$ distinct points, which we can sort in $\mathcal{O}(n \log n)$, so at most $2n + 1$ intervals per axis. It suffices to consider open intervals. We select one point from at most $(2n+1)(2n+1) = \mathcal{O}(n^2)$ regions (which we can index in a 2D array), and check in $\mathcal{O}(1)$ per rectangle if the point is included ($\mathcal{O}(n)$ per point), storing one counter per region. Finally, we iterate over the array and compute the maximum. This yields $\mathcal{O}(n^3)$ overall.

(c) Since the reduction is to a Clique, part (a) does not show that BoxDepth is NP-complete. Had we shown that an NP-complete problem (such as Clique) reduces to a problem in P (such as BoxDepth), we could have drawn this conclusion.

[1]**Helly's Theorem**. *Let K be a family of at least $d+1$ convex sets in $\mathbb{R}^d$, and assume K is finite or that each member of K is compact. Then if each $d+1$ members of K have a common point, there is a point common to all members of K.* Definition outline weakly adapted from: https://en.wikipedia.org/wiki/Helly's_theorem.

2. By reducing from 3SAT, we demonstrate that this puzzle is NP-hard. Specifically, given an instance ϕ of 3SAT in n clauses and m variables, we proceed with a polynomial transformation of the board into a puzzle configuration as follows: for all indices i and j, with $1 \leq i \leq n$ and $1 \leq j \leq m$, we fill each board entry (i, j) with a "snowman" whenever the boolean variable x_j is part of the i-th clause of ϕ. By symmetry, we add a "snowflake" at position (i, j) whenever the negation $\overline{x}_j$ is part of the i-th clause of ϕ. Finally, we default to empty spaces if neither x_j nor its negation $\overline{x}_j$ are part of the i-th clause of ϕ.

Claim — We claim this puzzle configuration is solvable if and only if ϕ is satisfiable.

Proof. We include both directions in the proof outline below.

$\Longrightarrow$ (*Sufficient Condition*) — assume the puzzle can be solved. Then we consider a solution construction where for each column index j we make an assignment to x_j that is True if column j contains a "snowman," False if column j contains a "snowflake," and arbitrary values of True/False if column j is empty. That is, assign values to the variables so that the literal corresponding to the remaining pieces are all True. Each row still has at least one puzzle piece, so each clause of ϕ contains at least one True literal, so this construction makes ϕ = True. Hence, ϕ is satisfiable.

$\Longleftarrow$ (*Necessary Condition*) — suppose ϕ is satisfiable; consider an arbitrary satisfying assignment. For each index entry j remove "snowmen" from column j according to the value assigned to x_j: (1) if x_j = True, remove all "snowflakes" from column j; (2) if x_j = False remove all "snowmen" from column j. That is, remove the puzzle pieces that correspond to False literals. Now, because every variable appears in at least one clause, each column contains at least one True literal, and thus each row still contains at least one puzzle piece, and hence the puzzle is satisfiable.

□

We only need linear time to perform this reduction, since we can allocate a column to a new variable as we go. Hence the transformation is polynomial-time and the reduction is complete.