Solution.

(a) Let $f : \mathbb{Z} \to \mathbb{Z}$ be a convex function, for example $f(x) = x^2$. Suppose we have an array of integers X, we can define a mapping into 2-dimensional space according to $X[i] \to (X[i], f(X[i]))$. We can then solve the ConvexHull problem. f can be run in $O(1)$ time depending on the function. The order they appear in from left to right will be calculated by ConvexHull, so after the problem is solved, you will have an ordered set of vertices. This operation will be a linear time reduction from Sorting to ConvexHull.

(b) The Sorting problem is well known to have a lower bound of $\Omega(n \log n)$, so we will assume this is true and the Sorting problem cannot be reduced. Now, let's suppose that there exists an algorithm that can compute ConvexHull in $o(n \log n)$ time. This would imply that we, post-reduction from part (a), we can determine Sorting in $o(n \log n)$, plus the linear time for reduction. In other words, it implies we can solve Sorting in less than $n \log n$ time. However, this implies that Sorting can be run in $o(n \log n)$, contradicting our assumption of $\Omega(n \log n)$ as a lower bound.