Instructions. Do this quiz in partnership with exactly one other student. Write both your names at the top of this page. Discuss the answer to the question with each other, and then write your joint answer below the question. Use the back of the page if necessary. It is fine if you overhear what other students say, because you still need to decide if they are right or wrong. You have seven minutes.

Question. The greedy algorithm to find a tree representing the structure of a sentence is the following, where $n$ is the length of the sentence:

- First, consider all $n - 1$ pairs of consecutive words. Evaluate the reconstruction error for each pair. Select the pair with smallest error, and combine it.
- Next, consider the remaining feasible pairs, plus the new possible pairs on top of the first selected pair. Select the pair with smallest error among these candidate pairs, and combine it.
- Continue until there is only one possible choice to create the root node.

Let $d$ be the dimensionality of meaning vectors. In terms of $n$ and $d$, what is the big-O time complexity of this algorithm?

Answer. Computing the meaning vector for one pair involves multiplying the $d$ by $2d$ matrix $W$ and a concatenated vector of length $2d$. The time needed is $O(d^2)$. The complexity of computing the reconstructed meanings is the same. The complexity of computing the error for one pair is $O(d)$. There are $n - 1$ pairs at the lowest level, so the complexity of identifying which error is smallest error is $O(n)$. The total time needed at the lowest level is thus $O(nd^2)$.

As mentioned in the lecture notes, after one pair is combined, either one or two new pairs need to be evaluated, so only $O(d^2 + n)$ time is needed for each later choice. There are $n - 1$ choices in total, so the total time complexity is $O(nd^2 + n^2) = O(nd^2)$. The last equation assumes that $n$ is small compared to $d^2$, which it is in practice.