

Task 2 Part C

Limiting the occurrence of each letter to only once per word allows us to implement an algorithm with a significantly higher efficiency. The fundamental idea behind this new algorithm is that, for each prefix, if a valid spot on our board can be reached by multiple paths, then we only consider it once. This means that we are not needlessly checking the same board positions multiple times. We do this by performing depth-first search on our prefix tree, finding and marking all possible occurrences of our first character, then finding all positions which neighbour these that match the next character, and so on. This means we are finding all positions which are valid for a particular prefix, storing this in a matrix, and then using this result to extend our solution to the next character. Without the requirement that there only be unique characters in a word, there is the chance that we mark a position as 'visited' which we might need to use later to complete the rest of the word, making this approach impossible.

This implementation results in an upper bound on the time complexity of $O(mn)$, where m is the number of nodes in the prefix tree and n is the total number of elements in the board (i.e. the board is of size $\sqrt{n} \times \sqrt{n}$). This is much better than the complexity of part A which is $O(n * 8^n)$, with n defined as above.