**SWE Project 3**

**Code Review:**

Hey Joshua,

The overall code looks complete and is implemented correctly. It is easy to follow through the code to understand what it is trying to achieve. Also, the data structures used are efficient and work well with the problem that you are trying to solve. However, there is still few rooms for improvement:

**Structure**

* The code is not consistently formatted and styled:
  + The indentation of the code is varying especially for if-else statements.
  + The whitespaces between different function declarations are not consistent. It appears as if it was part of a single function.
  + Functions and variables follow both camel case and snake case (eg: `is\_prefix` and `wordSearch`). It is better to use consistent styling throughout the code.
* Some checks can be refactored into a single function.
  + There is a function to check if a grid is valid or not using `validGrid`. However, some checks such as checking if grid is passed, grid is of size NxN can be considered checking whether a grid is valid or not. Hence, these checks can be consolidated into the same function `validGrid`.
* The code logs the following `console.log('TEST: ' + grid);` which does not provide any insightful information to the user calling the code.
* `lowerCase` function is converting strings of two different data structures (2D array and 1D array). It is better to split it into two different functions.
* While we should check if a word length is at least three to be a valid word, it is preferred to declare the number as a global constant and references it later such as `MIN\_WORD\_LENGTH`. This allows us to easily update the code if the requirement changes for what the minimum length of the word should be. The reader can also know what the number is for.

**Documentation**

* While most functions' name is descriptive enough to explain what it is trying to achieve, there are some functions that do not explain what it is trying to do: `lowerCase` and `hashmap`. For these functions, no comment is provided as well to document what it is trying to achieve.
* Most comments simply state what the code is doing instead of explaining why were are doing it. For example, “//is it a prefix for any word already in hash” is a conditional but does not explain why we have the conditional for this. What is special about prefixes for a word that is in a hash?

**Variables**

* Variable declaration is inconsistent. `let` and `var` are being used to declare variables. Following the conventions of Javascript ES6, it is preferred to use `let` over `var`.
* Some variables can be declared as `const` since they will not be reassigned and can help catch bugs earlier where we mistakenly reassign these variables.
* such as:
  + `N` (line 10)
  + `setSolution` (line 30)
  + `hash` (line 31)
  + `used` (line 38)
  + `dict` (line 74)
  + `str` (line 78)
* Some variables are not declared properly: `reg\_ex`. Use `const`
* Some variable names are not meaningful such as `hash`, `dict`, `reg\_ex`.

Great job on the implementation and solving the problem!