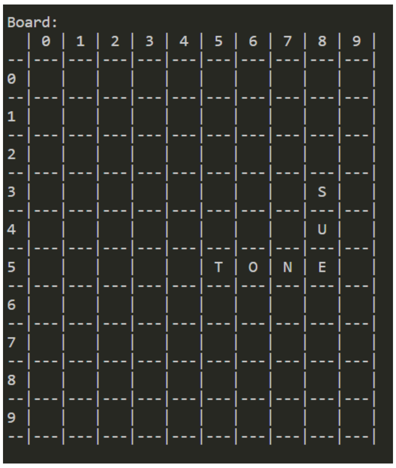
Brute Force Approach

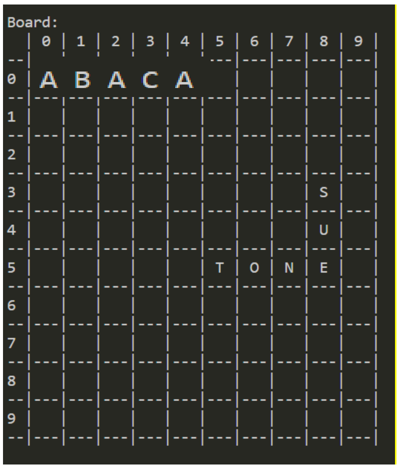
A brute force algorithm will go through every possibility in order to find the solution. Therefore, to implement this approach for the scrabble game, every word should be considered from the dictionary.txt and run through every possible location in the Board. Words and locations that are both **valid** will have their scores checked and if their score is higher than the current high score, it will supplant the previous best move as the current best move. This continues until all possibilities are considered.

A visual breakdown for the brute force algorithm will be covered for the 10x10 Board below.



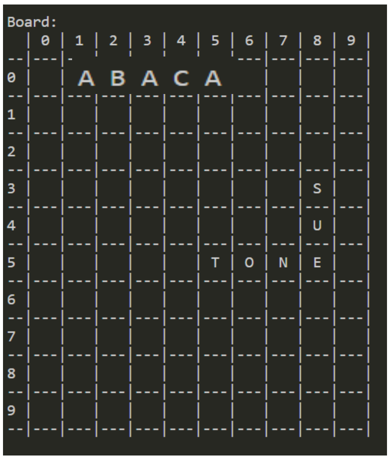
```
Board:
  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
--|---|---|---|---|---|---|---|---|---|---|
0 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
1 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
2 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
3 |   |   |   |   |   |   |   |   | S |   |
--|---|---|---|---|---|---|---|---|---|---|
4 |   |   |   |   |   |   |   |   | U |   |
--|---|---|---|---|---|---|---|---|---|---|
5 |   |   |   |   |   | T | O | N | E |   |
--|---|---|---|---|---|---|---|---|---|---|
6 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
7 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
8 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
9 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
```

The first word in dictionary.txt is "ABACA". "ABACA" will be first checked for the location "0:0:H", visualised below.



```
Board:
  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
--|---|---|---|---|---|---|---|---|---|---|
0 | A | B | A | C | A |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
1 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
2 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
3 |   |   |   |   |   |   |   |   | S |   |
--|---|---|---|---|---|---|---|---|---|---|
4 |   |   |   |   |   |   |   |   | U |   |
--|---|---|---|---|---|---|---|---|---|---|
5 |   |   |   |   |   | T | O | N | E |   |
--|---|---|---|---|---|---|---|---|---|---|
6 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
7 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
8 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
9 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
```

At this location, the word "ABACA" does not use any of the tiles already on the Board. Because it does not satisfy all of the conditions, this word at this location is not valid and hence it's score is not considered. The next location, 0:1:H, is considered (pictured below) and the one after that, all through until the end of the row. After the whole row has been scanned, the algorithm checks for the word at the next row (i.e. 1:0:H) and continues.



```
Board:
  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
--|---|---|---|---|---|---|---|---|---|---|
0 |   | A | B | A | C | A |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
1 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
2 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
3 |   |   |   |   |   |   |   |   | S |   |
--|---|---|---|---|---|---|---|---|---|---|
4 |   |   |   |   |   |   |   |   | U |   |
--|---|---|---|---|---|---|---|---|---|---|
5 |   |   |   |   |   | T | O | N | E |   |
--|---|---|---|---|---|---|---|---|---|---|
6 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
7 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
8 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
9 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
```

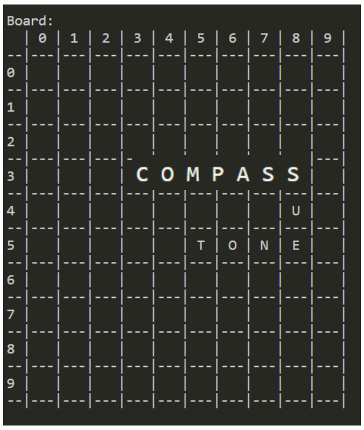
After all of the rows have been considered, the algorithm starts at 0:0:V and continues in the same pattern, with the next possible location at 0:1:V to 0:2:V to 0:3:V and so on.

Once all horizontal and vertical placements are considered for that word, the next word is considered with all horizontal and vertical placements, until all words in dictionary.txt are considered.

As long as the (word, location) pair is invalid, the algorithm will not consider the score of the move and will instead continue with a new location. However, at any point in the brute force algorithm where a word is indeed valid, the score will be computed.

The word below in consideration is “COMPASS” at the location 3:3:H. Assume that “COMPASS” can be made from the tiles. Because “COMPASS” uses a tile from the Board AND it can be made from the tiles, the score (calculated as 21) is compared to the current best word and best score. If the score of 21 is **higher** than the best score, then “COMPASS” is now the best word with a best score of 21. If the score of 21 is **lower** than the best score, then the previous best move will remain, and the algorithm will proceed.

Note that the brute force approach will still continue considering “COMPASS” at the remaining possible locations regardless of being the next best move or not, in order to cover every possibility.



```
Board:
  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
--|---|---|---|---|---|---|---|---|---|---|
0 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
1 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
2 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
3 |   |   |   | C | O | M | P | A | S | S |
--|---|---|---|---|---|---|---|---|---|---|
4 |   |   |   |   |   |   |   |   | U |   |
--|---|---|---|---|---|---|---|---|---|---|
5 |   |   |   |   |   | T | O | N | E |   |
--|---|---|---|---|---|---|---|---|---|---|
6 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
7 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
8 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
9 |   |   |   |   |   |   |   |   |   |   |
--|---|---|---|---|---|---|---|---|---|---|
```

Justification for the brute force approach

Through scanning through every possible location for every possible word in the dictionary.txt, the brute force approach has considered every possible solution available for the move. At a particular location, if the word does not satisfy all of the required conditions, such as not using a tile from the Board OR the word not being able to fit on the Board OR the word not being able to be made from the tiles, it will not be counted as a valid solution and hence only valid solutions are considered. The best valid solution will only replace the previous best if the score for the word at the location is higher than the previous best move. Overall, the brute force approach considers **all** possible solutions, considering only **valid** moves and having their scores compared against each other, which leads to the brute force approach always finding the best move by the end of the algorithm.