The Problem
Crosswords present several interesting problems from a computer science perspective, because they contain a large number of constraints. For example, a crossword puzzle is a grid of cells, each of which may either contain a letter or be empty. The program must satisfy several constraints for each cell: the letters must form words, the letters in each row and column must form words, and the words must intersect with other words.

The Data
Finding the right data sources was one of the most important aspects of this project, since my approach relies so heavily on data. I wrote a web scraper to gather this data from crossword solving sites, which I then combined with an existing list of clues from the New York Times crossword. The program works by generating lists of candidate answers for each clue, then reasoning for the configuration of answers that best satisfies all constraints.

The New York Times crossword is the basis of the project because it is the most well-known and most challenging, by general consensus among avid solvers. Consequently, the best crossword solving computer program should be able to solve the New York Times puzzle.

The Algorithm
The first module is a fuzzy search, which uses the edit distance metric to determine the similarity of two strings. This gives us a way to generate potential answers, even when the letters are in a different order. The second module is a letter matching algorithm, which uses a binary search to find the best possible answer. The third module is a letter matching algorithm, which uses a binary search to find the best possible answer.

1. Getting Answers

a. Modified Binary Search

This module is the first step in the program. It takes a clue and generates all possible answers for the clue, then ranks them based on the number of intersecting answers. The best possible answer is then used to generate all possible answers for the next clue.

b. Fuzzy Search

This module is the second step in the program. It takes a clue and generates all possible answers for the clue, then ranks them based on the number of intersecting answers. The best possible answer is then used to generate all possible answers for the next clue.

c. Letter Matching

This module is the third step in the program. It takes a clue and generates all possible answers for the clue, then ranks them based on the number of intersecting answers. The best possible answer is then used to generate all possible answers for the next clue.

2. Filling the Puzzle

Big O notation is the system of representing the upper limit of a function that is used to relate the efficiency of an algorithm to the size of the problem. In this case, the efficiency of an algorithm is related to the speed at which the program can solve the puzzle.

3. Results

Big O notation is the system of representing the upper limit of a function that is used to relate the efficiency of an algorithm to the size of the problem. In this case, the efficiency of an algorithm is related to the speed at which the program can solve the puzzle.

Limitations and Future Work

1. Puzzles Solved by My Program

2. The Problem

Crosswords present several interesting problems from a computer science perspective, because they contain a large number of constraints. For example, a crossword puzzle is a grid of cells, each of which may either contain a letter or be empty. The program must satisfy several constraints for each cell: the letters must form words, the letters in each row and column must form words, and the words must intersect with other words.

The Data
Finding the right data sources was one of the most important aspects of this project, since my approach relies so heavily on data. I wrote a web scraper to gather this data from crossword solving sites, which I then combined with an existing list of clues from the New York Times crossword. The program works by generating lists of candidate answers for each clue, then reasoning for the configuration of answers that best satisfies all constraints.

The New York Times crossword is the basis of the project because it is the most well-known and most challenging, by general consensus among avid solvers. Consequently, the best crossword solving computer program should be able to solve the New York Times puzzle.

The Algorithm
The first module is a fuzzy search, which uses the edit distance metric to determine the similarity of two strings. This gives us a way to generate potential answers, even when the letters are in a different order. The second module is a letter matching algorithm, which uses a binary search to find the best possible answer. The third module is a letter matching algorithm, which uses a binary search to find the best possible answer.

1. Getting Answers

a. Modified Binary Search

This module is the first step in the program. It takes a clue and generates all possible answers for the clue, then ranks them based on the number of intersecting answers. The best possible answer is then used to generate all possible answers for the next clue.

b. Fuzzy Search

This module is the second step in the program. It takes a clue and generates all possible answers for the clue, then ranks them based on the number of intersecting answers. The best possible answer is then used to generate all possible answers for the next clue.

c. Letter Matching

This module is the third step in the program. It takes a clue and generates all possible answers for the clue, then ranks them based on the number of intersecting answers. The best possible answer is then used to generate all possible answers for the next clue.

2. Filling the Puzzle

Big O notation is the system of representing the upper limit of a function that is used to relate the efficiency of an algorithm to the size of the problem. In this case, the efficiency of an algorithm is related to the speed at which the program can solve the puzzle.

3. Results

Big O notation is the system of representing the upper limit of a function that is used to relate the efficiency of an algorithm to the size of the problem. In this case, the efficiency of an algorithm is related to the speed at which the program can solve the puzzle.