

Laporan Tugas Kecil 1 IF2211 Strategi Algoritma 2021/2022

Oleh Muhammad Alif Putra Yasa, 13520135

Deskripsi Algoritma

Program ini dapat dibagi menjadi dua bagian, memparse puzzle, dan menyelesaikan puzzle. Oleh karena itu, kedua bagian akan dibahas secara terpisah.

Mem-parse Puzzle

Langkah 1 Baca seluruh file dan simpan dalam sebuah variabel bertipe `String`. Jika ada kesalahan di file, atau file tidak ada, *raise error*.

Langkah 2 Split file yang telah disimpan di variabel bertipe `String` dengan `'\n\n'` (*double enter*). Langkah ini akan memisahkan variabel tersebut menjadi dua bagian, bagian pertama yang berisi puzzle dan bagian kedua yang berisi daftar kata-kata.

Langkah 3 Dari bagian pertama hasil **Langkah 2**, lakukan dua split lagi menggunakan `'\n'` (*enter*) lalu dengan `' '` (*space*). Langkah ini akan menghasilkan array bertipe `String` dua dimensi. Puzzle yang telah di-parse ini kemudian disimpan di atribut `puzzle_matrix`.

Langkah 4 Dari bagian kedua hasil **Langkah 2**, split lagi dengan `'\n'`. Langkah ini akan menghasilkan array of `String`. Daftar kata-kata yang telah di-parse ini kemudian disimpan di variabel `puzzle_solutions`.

Langkah 5 Gunakan Puzzle yang telah di-parse di **Langkah 3** untuk menentukan dimensi puzzle, lalu menyimpannya di atribut `puzzle_rows` dan `puzzle_cols`. Puzzle kini telah selesai di-parse dan siap untuk di-solve.

Menyelesaikan Puzzle

Langkah 1 Siapkan array `String` dua dimensi yang berukuran sama dengan Puzzle yang ingin di-solve. Di langkah selanjutnya, array ini akan dipanggil `solvedPuzzle`. Isi seluruh sel `solvedPuzzle` dengan `-`. Kemudian siapkan variabel bertipe `Integer` untuk menyimpan jumlah perbandingan huruf. Di langkah selanjutnya, variabel ini akan dipanggil `compNumber`.

Langkah 2 Lakukan iterasi untuk tiap sel di puzzle. Setiap iterasi, panggil fungsi `cellCompare`. Fungsi ini akan mengisi `solvedPuzzle` dengan kata ditemukan sesuai dengan arahnya dan menambah jumlah `compNumber`. Detail mengenai `cellCompare` akan dibahas di langkah selanjutnya.

Langkah 3 Di `cellCompare`, akan dilakukan iterasi terhadap daftar kata-kata. Untuk setiap kata-kata, akan dipanggil fungsi `directionCompare`, yang akan melakukan perbandingan kata dari daftar kata-kata (selanjutnya akan disebut dengan *keyword*) dan *keyword* yang di-reverse (selanjutnya akan disebut dengan *reversed keyword*) dengan kata yang dimulai atau diakhiri di sel tersebut (selanjutnya akan disebut dengan *puzzleword*) berdasarkan empat arah, **KANAN**, **BAWAH**, **KANAN-BAWAH**, dan **KIRI-BAWAH**. Fungsi ini juga menghasilkan jumlah perbandingan yang telah dilakukan. Detail fungsi `directionCompare` akan dibahas di langkah selanjutnya.

Langkah 4 Di `directionCompare`, akan diukur panjang *keyword*. Berdasarkan arahnya, akan diambil `String` sepanjang *keyword*. Jika tidak cukup, akan diambil sebanyak mungkin. Kemudian, *puzzleword* ini akan dibandingkan dengan *keyword* dan *reversed keyword* menggunakan konstruktor `class StringCompare`. `Class StringCompare` berisi jumlah perbandingan dan hasil perbandingan kedua `String`. Pemanggilan konstruktor ini akan tetap dilakukan meski kedua kata tidak memiliki panjang yang sama. Detail mengenai `class` ini akan dibahas di langkah selanjutnya.

Langkah 5 Di `StringCompare`, akan diukur panjang kedua kata. Jika tidak sama, akan langsung menghasilkan **false**. Jika sama, akan dilakukan perbandingan perhuruf sampai selesai (kata sama) atau ada huruf yang berbeda. Jumlah perbandingan huruf disimpan di atribut `compNum` dan hasil perbandingan disimpan di atribut `compResult`. Hasil ini akan digunakan di fungsi `directionCompare`.

Langkah 6 Di `directionCompare`, hasil perbandingan *puzzleword* dengan *keyword* dan *reversed keyword* akan digunakan untuk mengisi `solvedPuzzle`. Jumlah perbandingan yang dilakukan akan dijumlahkan dengan perbandingan dengan arah lain, lalu jumlah ini akan di-*return* ke fungsi `cellCompare`.

Langkah 7 Di `cellCompare`, hasil dari `directionalCompare` akan dijumlahkan dengan `directionalCompare` menggunakan *keyword* lain. Jumlah dari seluruh hasil `directionalCompare` di sel tersebut akan di-*return* ke fungsi utama (**Langkah 2**).

Langkah 8 Kemudian akan dijumlahkan hasil `cellCompare` untuk seluruh sel yang ada di `puzzle_matrix`. Puzzle kini telah selesai, puzzle yang telah selesai dapat di-*display* menggunakan metode `display` beserta dengan jumlah perbandingan huruf yang telah dihitung sebelumnya.

Source Program

Bahasa Pemrograman yang digunakan untuk proyek ini adalah Java. Program terbagi menjadi tiga file, `Word_Puzzle.java`, `Puzzle_Solver.java`, dan `Main.java`.

Word_Puzzle.java

```
1  import java.io.IOException;
2  import java.nio.file.Files;
3  import java.nio.file.Path;
4
5  public class Word_Puzzle {
6      public Integer puzzle_rows;
7      public Integer puzzle_cols;
8      public String[][] puzzle_matrix;
9      public String[] puzzle_solutions;
10
11     public Word_Puzzle(Integer puzzleRows, Integer puzzleCols) {
12         puzzle_cols = puzzleCols;
13         puzzle_rows = puzzleRows;
14         puzzle_matrix = new String[puzzleRows][puzzleCols];
15         puzzle_solutions = null;
16
17         for (int rows = 0; rows < puzzleRows; rows++) {
18             for (int cols = 0; cols < puzzleCols; cols++) {
19                 puzzle_matrix[rows][cols] = "-";
20             }
21         }
22     }
23
24     public Word_Puzzle(String filepath) {
25         try {
26             String main_file[] = Files.readString(Path.of(filepath)).split("\\n\\n");
27             puzzle_matrix = parsePuzzleMatrix(main_file[0]);
28             puzzle_solutions = parsePuzzleSolutions(main_file[1]);
29             puzzle_rows = puzzle_matrix.length;
30             puzzle_cols = puzzle_matrix[0].length;
31         } catch (IOException e) {
32             System.out.printf("ERROR when reading %s\\n", filepath);
33             System.exit(0);
34         }
35     }
36 }
```

```

37     private static String[][] parsePuzzleMatrix(String puzzleMatrixString) {
38         String tempPuzzleMatrix[] = puzzleMatrixString.split("\\n");
39         String[][] puzzleMatrix = new String[tempPuzzleMatrix.length][
            tempPuzzleMatrix[0].split(" ").length];
40
41         for (int row = 0; row < tempPuzzleMatrix.length; row++) {
42             puzzleMatrix[row] = tempPuzzleMatrix[row].split(" ");
43         }
44         return puzzleMatrix;
45     }
46
47     private static String[] parsePuzzleSolutions(String puzzleSolutionsString)
48     {
49         return puzzleSolutionsString.split("\\n");
50     }
51
52     public void display() {
53         System.out.println("Word Puzzle: ");
54         for (int i = 0; i < puzzle_rows; i++) {
55             for (int j = 0; j < puzzle_cols; j++) {
56                 System.out.printf("%s ", puzzle_matrix[i][j]);
57             }
58             System.out.println("");
59         }
60
61         System.out.println("Word Solutions: ");
62         // for (int solNum = 0; solNum < puzzle_solutions.length; solNum++) {
63         //     System.out.println(puzzle_solutions[solNum]);
64         // }
65         int printIdx = 0;
66         while (printIdx < puzzle_solutions.length) {
67             do {
68                 System.out.printf("%s\t", puzzle_solutions[printIdx]);
69                 printIdx++;
70             } while (printIdx % 5 != 0 && printIdx < puzzle_solutions.length);
71             System.out.println();
72         }
73     }

```

Puzzle_Solver.java

```

1  import java.util.Objects;
2
3  public class Puzzle_Solver {
4
5      public String[][] solvedPuzzle;
6      public Integer compNumber;
7
8      public Puzzle_Solver(Word_Puzzle WP) {
9          compNumber = 0;
10         solvedPuzzle = emptyPuzzle(WP.puzzle_rows, WP.puzzle_cols);
11         for (int rowIdx = 0; rowIdx < WP.puzzle_rows; rowIdx++) {

```

```

12         for (int colIdx = 0; colIdx < WP.puzzle_cols; colIdx++) {
13             compNumber += cellCompare(WP, solvedPuzzle, rowIdx, colIdx);
14         }
15     }
16
17 }
18
19 private static String[][] emptyPuzzle(Integer rowNum, Integer colNum) {
20     String[][] resultPuzzle = new String[rowNum][colNum];
21     for (int rowIdx = 0; rowIdx < rowNum; rowIdx++) {
22         for (int colIdx = 0; colIdx < colNum; colIdx++) {
23             resultPuzzle[rowIdx][colIdx] = "-";
24         }
25     }
26     return resultPuzzle;
27
28 }
29
30 private static Integer min(Integer intA, Integer intB) {
31     return Math.min(intA, intB);
32 }
33
34 private static Integer min(Integer intA, Integer intB, Integer intC) {
35     return Math.min(intA, Math.min(intB, intC));
36 }
37
38 private static String reverseString(String str) {
39     String rString = "";
40     for (int idx = 0; idx < str.length(); idx++) {
41         rString = str.charAt(idx) + rString;
42     }
43     return rString;
44 }
45
46 private static String getHorizontalString(Word_Puzzle WP, Integer initX,
47     Integer initY, Integer charNum) {
48     String res = "";
49     for (int charIdxOffset = 0; charIdxOffset < min(charNum, WP.puzzle_cols
50         - initX); charIdxOffset++) {
51         res += WP.puzzle_matrix[initY][initX + charIdxOffset];
52     }
53     return res;
54 }
55
56 private static String getVerticalString(Word_Puzzle WP, Integer initX,
57     Integer initY, Integer charNum) {
58     String res = "";
59     for (int charIdxOffset = 0; charIdxOffset < min(charNum, WP.puzzle_rows
60         - initY); charIdxOffset++) {
61         res += WP.puzzle_matrix[initY + charIdxOffset][initX];
62     }
63     return res;
64 }
65
66 private static String getNegativeDiagonalString(Word_Puzzle WP, Integer
67     initX, Integer initY, Integer charNum) {

```

```

63     String res = "";
64     for (int charIdxOffset = 0; charIdxOffset < min(charNum, WP.puzzle_rows
        - initY,
65         WP.puzzle_cols - initX); charIdxOffset++) {
66         res += WP.puzzle_matrix[initY + charIdxOffset][initX +
            charIdxOffset];
67     }
68     return res;
69 }
70
71 private static String getPositiveDiagonalString(Word_Puzzle WP, Integer
    initX, Integer initY, Integer charNum) {
72     String res = "";
73     for (int charIdxOffset = 0; charIdxOffset < min(charNum, WP.puzzle_rows
        - initY,
74         initX); charIdxOffset++) {
75         res += WP.puzzle_matrix[initY + charIdxOffset][initX -
            charIdxOffset];
76     }
77     return res;
78 }
79
80 private static StringCompare horizontalCompare(Word_Puzzle WP, Integer
    initRow, Integer initCol, String toCompare,
81     String[][] resultPuzzle) {
82     String str = getHorizontalString(WP, initCol, initRow, toCompare.length
        ());
83
84     StringCompare normalComp = new StringCompare(str, toCompare);
85     StringCompare reverseComp = new StringCompare(str, reverseString(
        toCompare));
86
87     StringCompare compareResult = new StringCompare(normalComp.compNum +
        reverseComp.compNum,
88         normalComp.compResult || reverseComp.compResult);
89
90     if (compareResult.compResult) {
91         for (int idxOffset = 0; idxOffset < toCompare.length(); idxOffset
            ++){
92             resultPuzzle[initRow][initCol + idxOffset] = WP.puzzle_matrix[
                initRow][initCol + idxOffset];
93         }
94     }
95
96     return compareResult;
97 }
98
99 private static StringCompare verticalCompare(Word_Puzzle WP, Integer
    initRow, Integer initCol, String toCompare,
100     String[][] resultPuzzle) {
101     String str = getVerticalString(WP, initCol, initRow, toCompare.length()
        );
102
103     StringCompare normalComp = new StringCompare(str, toCompare);
104     StringCompare reverseComp = new StringCompare(str, reverseString(
        toCompare));

```

```

105
106     StringCompare compareResult = new StringCompare(normalComp.compNum +
107         reverseComp.compNum,
108         normalComp.compResult || reverseComp.compResult);
109
110     if (compareResult.compResult) {
111         for (int idxOffset = 0; idxOffset < toCompare.length(); idxOffset
112             ++){
113             resultPuzzle[initRow + idxOffset][initCol] = WP.puzzle_matrix[
114                 initRow + idxOffset][initCol];
115         }
116     }
117     return compareResult;
118 }
119
120 private static StringCompare negativeDiagonalCompare(Word_Puzzle WP,
121     Integer initRow, Integer initCol,
122     String toCompare,
123     String[][] resultPuzzle) {
124     String str = getNegativeDiagonalString(WP, initCol, initRow, toCompare.
125         length());
126
127     StringCompare normalComp = new StringCompare(str, toCompare);
128     StringCompare reverseComp = new StringCompare(str, reverseString(
129         toCompare));
130
131     StringCompare compareResult = new StringCompare(normalComp.compNum +
132         reverseComp.compNum,
133         normalComp.compResult || reverseComp.compResult);
134
135     if (compareResult.compResult) {
136         for (int idxOffset = 0; idxOffset < toCompare.length(); idxOffset
137             ++){
138             resultPuzzle[initRow + idxOffset][initCol + idxOffset] = WP.
139                 puzzle_matrix[initRow + idxOffset][initCol
140                     + idxOffset];
141         }
142     }
143     return compareResult;
144 }
145
146 private static StringCompare positiveDiagonalCompare(Word_Puzzle WP,
147     Integer initRow, Integer initCol,
148     String toCompare,
149     String[][] resultPuzzle) {
150     String str = getPositiveDiagonalString(WP, initCol, initRow, toCompare.
151         length());
152
153     StringCompare normalComp = new StringCompare(str, toCompare);
154     StringCompare reverseComp = new StringCompare(str, reverseString(
155         toCompare));
156
157     StringCompare compareResult = new StringCompare(normalComp.compNum +
158         reverseComp.compNum,

```

```

148         normalComp.compResult || reverseComp.compResult);
149
150     if (compareResult.compResult) {
151         for (int idxOffset = 0; idxOffset < toCompare.length(); idxOffset
152             ++){
153             resultPuzzle[initRow + idxOffset][initCol - idxOffset] = WP.
154                 puzzle_matrix[initRow + idxOffset][initCol
155                     - idxOffset];
156         }
157     }
158     return compareResult;
159 }
160
161 private static Integer cellCompare(Word_Puzzle originalWP, String[][]
162     resultPuzzle, Integer rowIdx,
163     Integer colIdx) {
164     Integer compNumber = 0;
165     for (int keywordIdx = 0; keywordIdx < originalWP.puzzle_solutions.
166         length; keywordIdx++){
167         String keyword = originalWP.puzzle_solutions[keywordIdx];
168         compNumber += horizontalCompare(originalWP, rowIdx, colIdx, keyword
169             , resultPuzzle).compNum;
170         compNumber += verticalCompare(originalWP, rowIdx, colIdx, keyword,
171             resultPuzzle).compNum;
172         compNumber += negativeDiagonalCompare(originalWP, rowIdx, colIdx,
173             keyword, resultPuzzle).compNum;
174         compNumber += positiveDiagonalCompare(originalWP, rowIdx, colIdx,
175             keyword, resultPuzzle).compNum;
176     }
177     return compNumber;
178 }
179
180 public void display() {
181     System.out.println("Solved Puzzle: ");
182     for (int rowNum = 0; rowNum < solvedPuzzle.length; rowNum++) {
183         for (int colNum = 0; colNum < solvedPuzzle[0].length; colNum++) {
184             System.out.printf("%s ", solvedPuzzle[rowNum][colNum]);
185         }
186         System.out.println("");
187     }
188     System.out.printf("Total Character Comparison: %d Comparison(s)\n",
189         compNumber);
190 }
191
192 class StringCompare {
193     public Integer compNum;
194     public Boolean compResult;
195
196     public StringCompare(String strA, String strB) {
197         compNum = 0;
198         compResult = Objects.equals(strA.length(), strB.length());
199     }
200 }

```



```

195         Integer compMax = Math.min(strA.length(), strB.length());
196         while (compResult && compNum < compMax) {
197             compResult = Objects.equals(strA.charAt(compNum), strB.charAt(
198                 compNum));
199             compNum++;
200         }
201
202         public StringCompare(Integer comparisonNum, Boolean comparisonResult) {
203             compNum = comparisonNum;
204             compResult = comparisonResult;
205         }
206     }

```

Main.java

```

1  import java.util.Objects;
2  import java.util.Scanner;
3
4  public class Main {
5      public static void main(String[] args) {
6
7          Scanner consoleInput = new Scanner(System.in);
8
9          String filePath;
10         String queryAnswer;
11         Boolean repeat;
12
13         do {
14
15             System.out.printf("Input Filepath: ");
16             filePath = consoleInput.nextLine();
17
18             // Load File
19             Word_Puzzle wp = new Word_Puzzle(filePath);
20
21             // Display Puzzle Info to Console
22             wp.display();
23             System.out.println();
24             System.out.printf("Solving %d x %d Word Puzzle...\n", wp.
25                 puzzle_rows, wp.puzzle_cols);
26
27             // Timer Start
28             long startTime = System.nanoTime();
29
30             // Solve Puzzle
31             Puzzle_Solver solvedPuzzle = new Puzzle_Solver(wp);
32
33             // Timer Ends
34             long endTime = System.nanoTime();
35
36             // Display Result
37             float duration = (endTime - startTime) / 1000000;

```

```

37         System.out.printf("Solved in %.2f ms\n", duration);
38         solvedPuzzle.display();
39
40         do {
41             System.out.printf("Solve another puzzle? (Y|N): ");
42             queryAnswer = consoleInput.nextLine();
43         } while (!Objects.equals(queryAnswer, "Y") && !Objects.equals(
                queryAnswer, "N"));
44
45         repeat = Objects.equals(queryAnswer, "Y");
46     } while (repeat);
47
48     System.out.println("Thank You! Exiting Program...");
49     consoleInput.close();
50 }
51 }

```

Tests

Test Case 1 (Small Puzzle)

File: test/Small1.txt

```

[malifpy@fedora word-search-puzzle-bruteforce]$ java -jar Tucill_13520135.jar
Input Filepath: test/Small1.txt
Word Puzzle:
J S O L U T I S
S U N A R U U A
N E P T U N E T
S O N I E I S U
R C E V T R E R
A H T R A E S N
M M E R C U R Y
Word Solutions:
EARTH   JUPITER MARS   MERCURY NEPTUNE
SATURN  URANUS  VENUS

Solving 7 x 8 Word Puzzle...
Solved in 70.00 ms
Solved Puzzle:
J - - - - - S
S U N A R U U A
N E P T U N E T
S - - I E - - U
R - - V T - - R
A H T R A E - N
M M E R C U R Y
Total Character Comparison: 947 Comparison(s)
Solve another puzzle? (Y|N): Y

```

Test Case 2 (Small Puzzle)

File: `test/Small2.txt`

Input

```
Input Filepath: test/Small2.txt
Word Puzzle:
T M T N Q M M Y T W N V X C D
E K Q E A R T R U S A Z R O D
X T S O Q K V R R S R F Q Q B
A U T T Z U Y P Y M U H Y W D
S F E D O X I S Y T Z Z G C U
F R W S T R T L N Y U X T M W
N W A D V O R J A O S H W C V
Q O R V R L Q U P B O A P U R
P O D M Z G U D S R C U Q A J
E K E D S J E X N S I O I D C
I Y A Q E W E S B S U S M Z H
E M U R N O T O B G Z S G H P
Y M I H F P T A C H A N K A F
H W S N O F I D D O T X N H T
S X R S Q O H L E V R D Z B J
Word Solutions:
STEWART STORMEYE SURTR SUSSURRO SUZURAN
SWIRE TACHANKA TEQUILA TEXAS THORNS
TODDIFONS
```

Output

```

Solving 15 x 15 Word Puzzle...
Solved in 52.00 ms
Solved Puzzle:
T - T - - - - - N - - - -
E - - E - R T R U S A - - - -
X - S - Q - - - - R - - - -
A - T - - U - - - - U - - - -
S - E - O - I - - - - Z - - - -
- - W - - R - L - - U - T - -
- - A - - - R - A - S H - - -
- - R - - - - U - - O - - - -
- - D - - - - - S R - - - - -
- - - - - - - N S - - - - -
- - - - - - - S - - U - - - -
- - - - - - - - - S - - - -
- - - - - - T A C H A N K A -
- - S N O F I D D O T - - - -
- - - - - - - - - - - - -
Total Character Comparison: 10393 Comparison(s)
Solve another puzzle? (Y|N): Y

```

Test Case 3 (Small Puzzle)

File: `test/Small3.txt`

Input

Input Filepath: test/Small3.txt

Word Puzzle:

```
E F O O R P S W E P O D F G Q
I T P A E R E G N E V A C S Y
J I Z M E A F F E O H P P P T
F D U G N M U D C I T R A D M
O L N R B A S I S P O L I T P
P A O S S N R B B V J Y G M X
R P U A I I G O E O O Q R A N
E M X V P X T N B H C V O E Z
A S V A C O C N I E V L B R O
H U M G W E P S O D R U I T O
M J S E S S U U A M D T N S A
S A I L E A C H K R S U A E A
H E P O D E N C O A I O P R G
J D F R J U Q Y X S R A R U A
R E E D A S O R F O B K Z P S
```

Word Solutions:

PLUME	PODenco	POPUKAR	PRAMANIX	PROVENCE
PTILOPSIS		PUDDING	PURESTREAM	RANGERS REED
ROBERTA	ROBIN	ROPE	ROSA	ROSMONTIS
SAGA	SAILEACH		SARIA	SAVAGE SCAVENGER
SCENE				

Output

```

Solving 15 x 15 Word Puzzle...
Solved in 60.00 ms
Solved Puzzle:
- - - - - P - - E - - - - -
- - - - - R E G N E V A C S -
- - - - - A - - E - - - P - -
- - - - - M - - C - - R - - -
- - - - - A S I S P O L I T P
- - - S S N R - - V - - - M -
- - - A - I G O E - - - R A -
- - - V P X T N B - - - O E -
- - - A - O C N I E - - B R -
- - - G - E P S O D R - I T -
- - - E - - - U A M D T N S -
S A I L E A C H K R S U A E A
- - P O D E N C O A I O P R G
- - - - - - - - - R A R U A
R E E D A S O R - - - - - P S
Total Character Comparison: 20738 Comparison(s)
Solve another puzzle? (Y|N): N
Thank You! Exiting Program...

```

Test Case 4 (Medium Puzzle)

File: `test/Medium1.txt`

Input

```

[malifpy@fedora word-search-puzzle-bruteforce]$ java -jar Tucill_13520135.jar
Input Filepath: test/Medium1.txt
Word Puzzle:
F E D T H C D N S J V W Q T E G F Y G F
I A X W U I Y K L I V F F M U V Y K I N
N K D T D O G M V A W Z Z Y B Q C R X E
D E N S K N B W X D K W R H U R S N Z J
O E P N I A P A E F H Q K Y T T H I Z I
F K A T S O V Z G M K I W U C X W H V C
G C D C A I O P Z P O D O W N B V H A S
N A H M H U B T Z I L C Z S X W R C A Y
H S N O E P I R G N L P Y T R S I Z O X
J Q M W R B C W G Y A J B N G C F M W R
R V J Z F D Y L W Y M O Q L Y L V F B Z
L Y C T R T M R D C Q C Q A T M M F G H
Q H N Q O Z U N C J S C P O Y B H W Y J
E T W U B N A S H Q S I W W G C B Z A P
B C I V B R D Z H E Q Z P B M T B K D L
M E O P W W N R A C R H W P L O V Z V C
S O E Y G L G Q F G T W P A R E O Z A M
H B R N J P S M S W E Q C C J H P L J F
N A C F C C I U V H G P X L E N L A K L
N G I L N O R O F J J D J J Z L F E F M
Word Solutions:
ABOUT    ALL      AND      ARE      BEEN
BUT       CALL    CAN      COME    DAY
DID       DOWN    EACH     FIND    FIRST
FOR       FROM    GET      HAD     HAS
HER

```

Output

```

Solving 20 x 20 Word Puzzle...
Solved in 121.00 ms
Solved Puzzle:
F - - T - - D - - - - - T E G - - - F
I - - - U I - - - - - - - - - I -
N - - - D O - - - - - - - B - - R - -
D - - - - B - - - - - - U - S - - -
- E - N - - - A E - - - - T T - - - -
- - A - S - - - - M - - - - - - -
- C - C A - - - - - O D O W N - - H A S
- - - - H - - - - - L C - - - - -
- - - - E - - - - - L - - - - -
- - - - R - - - - - A - - - - -
- - - - - - - - - - - - - - -
- - - - - - - D - - - - - - -
- - - - - - - N - - - - - - Y -
- - - - - A - H - - - - - - A -
B - - - - - - - E - - - - - D -
M E - - - - - - - R - - - - - C
- O E - - - - - - T - - A R E - - A -
- - R N - - - - - - E - - - - - L - -
N A C F - - - - - G - - - - - L - - -
- - - - - R O F - - - - - - - -
Total Character Comparison: 56622 Comparison(s)
Solve another puzzle? (Y|N): Y

```

Test Case 5 (Medium Puzzle)

File: `test/Medium2.txt`

Input


```

Solve another puzzle? (Y|N): Y
Input Filepath: test/Medium2.txt
Word Puzzle:
W P D D H B C K R D X T H E Y E Q T R U
G H U V T S I O Y U T D B S I F W K L O
O H A O I G X W B F I T D X R O E O H W
T J E T W T X W D G M A U N S H W L A I
K Q T T Y J H R V C E L L W Y A O V W X
W M Q D E I O L G F G Y V P A T W E R E
W E S U C W F R L J S U Z Y J Z H Q B B
V X I H Q Q Z E A I Z I C C Y H H I F S
N C C W C G B T Y Q W I Z T F E U J S W
F B I C V X N A O L G P O W W O Z N V Y
W L X U Z O Q W U Q E F J U H V R O T D
Z R F Q B S P G J W C K Z M E J O G J Q
I R I Q J M F R H Q Q K C Y N W O U L D
W S D T E P T E V T Z Q A Z Y B O Y F P
D O T S E Y U H T Q R W Z W I D R N P Q
V A E I O D M N F S I Q I R M H R I K T
N H K U J E F G G T Q D A M F N P W S M
T J R K I B M S T I N A Q V B O P X N C
C G F E H Z Y B B M P W M L J N N B W T
Z B U C F N V Z B R D U Y M A H U L F N
Word Solutions:
THESE      THEY      THIS      TIME      TWO
USE         WAS       WATER     WAY       WERE
WHAT       WHEN      WHICH     WHO       WILL
WITH       WORD      WOULD     WRITE     YOU
YOUR

```

Output

```

Solving 20 x 20 Word Puzzle...
Solved in 46.00 ms
Solved Puzzle:
W - - - H - - - - - T H E Y - - T - -
- H - - T - - - - - T - - - - - W - - -
- - A - I - - - - - I - - - - - O - O H W
- - - T W - - W D - M - - - S - - - - -
- - - - - H R - - E - - - - - A - - - - -
- - - - - I O L - - - - - T W E R E
- E S U C W - R L - - - - - H - - - - -
- - - H - - - E - I - - - - - I - - -
- - - - - T Y - W - - - - - S - - -
- - - - - A O - - - - - W - - - - -
W - - - - - W U - - - - - H - - - - -
- R - - - - - - - - - - - E - - - - -
- - I - - - - - - - - - - Y N W O U L D
- - - T - - - - - A - - - - -
- - - E Y - - - - - W - - - - -
- - - O - - - - - - - - - - -
- - - U - - - - - - - - - - -
- - R - - - - - - - - - - -
- - - - - - - - - - -
- - - - - - - - - - -
Total Character Comparison: 55382 Comparison(s)
Solve another puzzle? (Y|N): Y

```

Test Case 6 (Medium Puzzle)

File: [test/Medium3.txt](#)

Input

```

Solve another puzzle? (Y|N): Y
Input Filepath: test/Medium3.txt
Word Puzzle:
P J V J C D M L N M C T P L L P A F M P
Z Y L H Z U G X O P U B E U E R G K U H
K A C N I W Z A R T I I Y J R R B E I O
B M G C R M L U O G Y L L Z A H L P S S
W H L W L U K X B L R A T L U V O N E P
J A A F M H H X A M K E I V Y T R O N H
C G T I X E H T L Q U W C J A R C T G O
R P N T N E G O R D Y H J S F U E M A R
H U O E L M N P Q V A E S D J F T B M U
M H C I S O U J L O H I M N E L E V E S
K Y U H E A Y I V K U W L C M U F U L C
O M J N L H E L H M F C U G F S O K Z O
F E Q M L O A Y K T S A I J L X O M J W
N O G R A R R V M O I M X F U I C X N Z
C E I Y L O E I M N Z L A G O N L S H H
C A R B O N X E N U S N I T R O G E N V
M Y U D R S X Y T E I E H X I P V I L T
N O C I L I S K G C H D O F N R P O I J
B F X N O K H T S E V I O G E V N L B H
S Q P U D Q O L A N N V O S G J U C B Q
Word Solutions:
ALUMINUM          ARGON      BERYLLIUM      BORON    CALCIUM
CARBON    CHLORINE      FLUORINE      HELIUM    HYDROGEN
LITHIUM  MAGNESIUM      NEON        NITROGEN  OXYGEN
PHOSPHORUS      POTASSIUM      SILICON  SODIUM    SULFUR

```

Output

```

Solving 20 x 20 Word Puzzle...
Solved in 24.00 ms
Solved Puzzle:
- - - - - N M - - - - - M P
- - - - - O - U - - - - - U H
- - - - - R - - I - - - - - I O
- - - - - O - - - L - - - - P S S
- - - - - B - - - - L - - O - E P
- - - - - - - - - - Y T - - N H
- - - - - H - - - - - A R - - G O
- - - - N E G O R D Y H - S - U E - A R
- - - - L M N - - - - S - - F - B M U
- - C I - O U - - - - I - - - L - - S
- - U H E - - I - - U - - - U - - -
- M - N L - - H M - - - - F S - - -
- - - - - O - - - T - - - - L - - -
N O G R A - R - - - I - - - U - - -
- - - - - O - I M - - L - - O - - -
C A R B O N X - N U - N I T R O G E N -
- - - - - Y - E I - - - I - - -
N O C I L I S - G - - D - - N - - -
- - - - - E - - O - E - - -
- - - - - N - - S - - -
Total Character Comparison: 39396 Comparison(s)
Solve another puzzle? (Y|N): N
Thank You! Exiting Program...

```

Test Case 7 (Big Puzzle)

File: `test/Big1.txt`

Input

```
[malifpy@fedora word-search-puzzle-bruteforce]$ java -jar Tucil1_13520135.jar
Input Filepath: test/Big1.txt
Word Puzzle:
L Q A D Q X F E G O G O J U E U W I M I R I Z J A R W W J A W I
E N Y I S F H S N A V C A S L K G S B L R L R X M A D H X A I L
I C J V E S Y O Z J S E Y F E I R A Y R C L L F A M Y F K H C E
B H N X C Y X R L C A J E L W C K L Q K Q A H W H V Q P X Q W E
P E G F V N J I N F B R S Z A K I Z P U V C Q B C Z B G H T V E
E S G K V O V K H U W U V I M Y I R G A C K O C A R N S M P W G
T N T T E G X A V F T R U Z X R O O O H S H D U A I L A P Y I J
M S Q M N I H Y F X U Z V U G I D W B H T I O N H Y Z Z R I J U
Q P A Q F D N F L A W R F Q R S L V Y E A I K C Y F B W A N Y A
R I J C F Y H Z Z Q I P C P C W J Z D V E X C Q O B M T V U S X
W R O V U Y H K C J X U R K Z L H A M Y M X F L T M W S X Q J T
W Z Q V T T I V E Z K C Q V U U W Y Y C N K K A Q M R X Q E V V
X Q Q Z P A E S Q T L Q K N L O N A G J S R J I I I M R Z Z N X
N C U E K E I T F M I Z A F O G K M K W G U N D O N X P M V I Y
G C U A J R B U B N L N Y S T I G E E B K L R O W K F N Y P R T
D T C Y Q U A O A R A I K J G M Y X Z Z G E J D S L Q K U Y O B
I U H W N S G R L W C O K R N X B T N L J S V U A V Q W W A Y S
J J Q D D C S C U N X A G V W N V C U S D Y O R G C G R N S O J
D N Z L C H L O E G N U F X C I N Z X H G C E R R O X S Z X K E
T M L E O K P W X A Y M J K C H S A C O S B P Z Y C M J V Y T O
H T K A N W K A T I V O Q L S J R M I L I Y Q T O O L A M Y J T
H F N L P O N A S L D V J J P W E A I T P B Q Q A U F W Q P Q C
N F L B N L R B H Y S K B C C K H G S A T T Q S L E A B S Q Q Y
B A W W G A U O R F A P V C V M S V A G S V C J C A F J B R W F
M D T I O F I S K U B H L N F E J W I N A G P I P D M M I O M M
G E H O W R J K P Z J L K W R T F H S U Q K I K V L F I J D K U
F B Q K B G P E O Y A D K E O Y J C T A I X R S W M N F W L V S
G J J K J W B G K H E G C T S C J G H Q F J Y N K O N O A O H W
V S E I U G Q K U O X U Z N Q P Y Q A Y A O S N R L S Y K I N E
D V K I G T S H Y C O L G O R Y O C G W D Q I K U B U F E O G S
D K Y Y T U W L K P C U R M O H E O M Y P J U S R H T C M O X P
K Y C B Z S X U R W B I Y N G H H W K A T H P A B L B G U X E J
Word Solutions:
AKIROSE AME ANYA AQUA AYAME
AZKI BAELS BOTAN CALLI CERES
CHLOE CHOCO COCO FLARE FUBUKI
GURA HAACHAMA INA IOFI IROHA
IRYS KANATA KIARA KORONE KOYORI
KRONII LAMY LAPLUS LUI LUNA
```

Output

Solving 32 x 32 Word Puzzle...

Solved in 165.00 ms

Solved Puzzle:

```
- - - - - E - - - - - I - - A - - - - -
- - - - - S - - - - - L - - M - - - - -
- - - - - O - - - - - L - - A - - - - -
- - - - - R - - - - - A - - H - - - - -
- - - - - I - - - - - I - - C - - C - - - - -
- - - - - K - - - - - R - - C - - A - - - - -
- - - E - - A - - - - - O - - H - - A - - - - -
- - M - - - - - - - - - - H - - O - H - - - - -
- - A - - - - - A - - - - - A - - C - - - - A N Y A
- - - - - Z - - - - - - - - - - O - - - - -
- - - - - K - - - - - L - A - - - - - - - - -
- - - - - I - - - - - U - Y - - - - - - - - -
- - - - - - - - - - N - - A - - - - - - - - -
- - - - - - - I - A - - - - M - - - - - - - - I -
- - - - - - - N - - - - - E - - - - - - - F - - - R -
- - - - - A - A R A I K - - - - - - - L - - - - O -
- - - - - R - - - - - K - - - - - - - A - - - - Y -
- - - - - U - - A - - - - - - - R - C - - - - O -
- - - - - C H L O E G N - - - - - - - E - - O - - - - K -
- - - L E - - - - - A - - - - - - - - - C - - - - -
- - - A N - - - - - T - - - - - - - - - O L A M Y - -
- - - P O - A - - - - - - - - - - - - - - - - -
N - - - - L R - - - - - - - - - - - S L E A B - - - -
- - A - - - - U O - - - - - S - - - - - - - - - - -
- - T I O F I S K - - - - - E - - I N A - - - - - I - -
- - - O - - - - - - - R - - - - - I - - - - - I - -
- - - - B - - - - - E - - - - - R - - - - - N - - - -
- - - - - - - C - - - - - - - Y - - O - - - - -
- - - - - - - - - - - - - A - S - R - - - - -
- - - - - - - L - - - - - - - Q I K U B U F - - - -
- - - - - - - U - - - - - - - U - - - - - - - -
- - - - - - - I - - - - - - - A - - - - - - - -
```

Total Character Comparison: 211658 Comparison(s)

Test Case 8 (Big Puzzle)

File: `test/Big2.txt`

Input

Solve another puzzle? (Y|N): Y

Input Filepath: test/Big2.txt

Word Puzzle:

```
R I K F E N R O E G X S G A Q W B E V Z W A K V Z T I B A Z L Y
G L C T H B Q W B S K H O R G X R N V G X L K J W X E Q M K Q A
H E M U T Q M W R E M N K E B N E S N A L H Z A M I O L B B O N
H H A D N A C H I E L O B N G T I F V N V V J S F G W D R K H K
Y T M L I S Y L S H R Z Q E K G I I C Y E H M Q Q U T P I S F A
T K W C S X C F K M O Y R M N C V W X P I T B J G E Y H E W N N
Y U N F B A W Z I I A E H U D X O M D A Z E C A T K X U L N G E
I J A T A F Q H P A J A B N Y A C L N C W N U S G G J Z Z T J B
E A S Y K F C Q M T L C Y L E X H R H L C S O I Z P C F S N Y N
A Y T X G Y F T O O Y I K Y E E R I Z S V T E G R T I W V Y S W
X N F M K Z D J R J A C A X A W G Q J R A F U E H X V P O P L X
N R I E Q E T P O X T S U M L Q E T K T H Z A P B W U P E Y U W
Y L E L N H O A P R I L A S P Y R F C S V W Z J L K L R R A J T
I X X O E R O P A J E J I G D P D Z L V P O B M B N E W G C V J
I E N K D G M S I M J J G L A I Z Z E H S A L Q Z P B Q Y E W T
K U B D C D N Z S F J C G M W H K L S M F K R G V H E W Y D F I
D V I E F S P A E G P T I V P A A K N V X V E P K S A K G K C Z
F C C A I Y H W T H A Y Y S T Q L U A G W J Q Y I B G I M A R E
A R H C R U X I S D A H K S W R X O X J Y Z Y Z M U L Z M T N F
K X T C V C H R A I C G O B C P Y B I X S Z V N H Z E Z Y S Y K
N C C L Z M H B Z P Q A H A W L R F D W J O U W D Z P F S B Q R
J F L S Y O J E T F H K Y V N R B R H Y F H U Z L N O N Z K O F
A N A E R D N A T K C A Z X Z E F O E L N R S V R F L J U Y D L
A G V E U H Q M Q T R D A F A A Y D L A Q A G Y N D T Q X P O S
P D U S C Y S U A O O G G N N Q X C S K L P M O Z M Y O H G M G
V S A F S C U A R T U W S P P S H B F D Y M Z P Z E C A G N M B
E J M A T I B U D D N T C D P A E X Y Y B L N U D R F L V K D Y
V V Y L C B A Y M E A V M X X S M B K H I P A I M W H S Y N E W
M Q J Q E J D C Y L J S R P T A N C L A C C L C S L J U Y K L M
W F Q T V C Y R K I W A R O Y B H U W I A V N O W A M F J U A A
R E T N U H E E B I V O S O F P G W S L N U V Y G W S S H Z J V
L B E E O D H Q S I X A W U I H A Y E R S C A R P E I Y E P S T
```

Word Solutions:

AAK	ABSINTHE	ACIDDROP	ADNACHIEL	AKAFUYU
AMBRIEL	AMIYA	ANDREANA	ANGELINA	ANSEL
AOSTA	APRIL	ARCHETTO	ARENE	ASBESTOS
ASH	ASHLOCK	ASTESIA	AURORA	AYERSCARPE
BAGPIPE	BEAGLE	BEANSTALK	BEEHUNTER	BEESWAX
BENA				

Output

Solving 32 x 32 Word Puzzle...

Solved in 79.00 ms

Solved Puzzle:

```
- - - - E - - - - - A - - - - - A - - - - - A - - - -  
- - - - H - - - - - R - - - - - K - - - - - M - - - -  
- - - - T - - - - - E - - - - - A - - - - - B - - - -  
- - A D N A C H I E L - - N - - - - - F - - - - - R - - - -  
- - - - I - - - - - E K - - - - - U - - - - - I - - - - A  
- - - - S - - - - - C - - X - - - B - - - - - Y - E - - - N  
- - - - B - - - - - O - - A - - - A - - - - - U L - - - E  
- - - - A - - - - - L - - W - - - G - - - - - - - - B  
- - - - - - - - - - - H - - S - - - P - - - - - - - -  
A - - - - - - - - - - - S - - E - - - I - - - - - - - -  
- N - - - - - - - - - - - A - - E - - - P - - - - - - - -  
- - I - - - - - - - - - - - B - - - - - E - - - - - - - -  
- - - L - - - A P R I L - - - - - - - - - - - - - - - - - -  
- - - - E - - - A - - - - - L - - - - - - - - - - - - - -  
- - - - - G - - I - - - - A - - - E H S A - - - - - B - - - -  
- - - - - N - S - - - - M - - - - S - - - - - - - - E - - - -  
- - - - - A E - - - I - - A A K N - - - - - - - - A - - - -  
- - - A - - - T - - Y - - T - - - A - - - - - - - - G - - - -  
- - - - R - - - S - A - - S - - - - - - - - - - L - - - - -  
- - - - - C - - A - - - O - - - - - - - - - - - - E - - - -  
- - - - - H - - - - A - - - - - - - - - - - - - - - - - -  
- - - - - E - - - - - B - - - - - - - - - - - - - - - - - -  
A N A E R D N A T - - A - - - E - - - - - - - - - - - - - -  
- - - - - H - - - T R - - - A - - - - A - - - - - - - - - -  
- - - - - S - - O O - - N - - - - S - - - - - - - - - - - -  
- - - - - A R - - - S - - - B - - - - - - - - - - - - - -  
- - - - - U - - - T - - - E - - - - - - - - - - - - - -  
- - - - - A - - - A - - - S - - K - - - - - - - - - - - -  
- - - - - - - - L - - - - T - - - A - - - - - - - - - - - -  
- - - - - - - - K - - - - O - - - - - A - - - - - - - - - -  
R E T N U H E E B - - - S - - - - - - - - - - - - - - - -  
- - - - - - - - - - - - - - A Y E R S C A R P E - - - - -
```

Total Character Comparison: 168105 Comparison(s)

Solve another puzzle? (Y|N): N

Thank You! Exiting Program...

Links and Check Lists

GitHub Link <https://github.com/malifpy/word-search-puzzle-bruteforce>

Poin	Ya	Tidak
Program berhasil dikompilasi tanpa kesalahan (<i>no syntax error</i>)	X	
Program berhasil <i>running</i>	X	
Program dapat membaca file masukan dan menuliskan luaran.	X	
Program berhasil menemukan semua kata di dalam puzzle.	X	