

# **Penyelesaian Word Search Puzzle dengan Menggunakan Algoritma Brute Force**

Laporan Tugas Kecil 1 IF2211 Strategi Algoritma

Semester II Tahun 2021/2022



Disusun oleh:

Grace Claudia 13520078

## I. PENJELASAN ALGORITMA

### A. Algoritma Brute Force

Brute Force merupakan salah satu pendekatan penyelesaian masalah yang memiliki ciri-ciri khusus yaitu penyelesaian yang bersifat **straightforward**. Penyelesaian ini dilakukan dengan **sangat sederhana** dengan mengorbankan jumlah langkah penyelesaian masalah. Dalam program, pengaruh penggunaan algoritma Brute Force adalah durasi pengeksekusian program. Dengan pertimbangan ini, brute force kerap kali kita temukan hanya pada **testcase yang kecil**. Program dengan metode Brute Force cenderung memakan waktu yang sangat lama untuk *testcase* yang besar.

### B. Algoritma Program secara General

1. Pembacaan input teks yang puzzlenya akan diubah kedalam bentuk matrix NxN bernama **puzzle**, kata yang dicari dikumpulkan dalam sebuah vector bertipe string dengan nama vector **points**
2. Untuk pengubahan puzzle ke matrix, akan dilakukan **iterasi satu-satu terhadap line** yang ada pada file dan berakhir jika line dalam file tersebut **tidak memiliki elemen (“”)** yang berarti akan dilanjutkan dengan proses pemasukan kata yang dicari ke vector
3. Untuk pemasukan kata ke vector, akan **dipisahkan terlebih dahulu puzzle** yang berada di atas text file dengan penggunaan loop while, lalu **sampai end of file, kata-kata yang dicari akan di push** ke vector.
2. Mengiterate kata-kata yang terdapat dalam vector dan satu-satu dicari sesuai urutan kemunculan dalam file dengan algoritma brute force lalu mengembalikan vector berisi struct point yang berisi koordinat letak x dan y huruf pada kata yang ditemukan
3. Mendisplay character yang ditunjuk pada vector berisi point, selain itu diisi “-”

### C. Penjelasan langkah-langkah algoritma brute force pada program

Penggunaan algoritma Brute Force terdapat pada fungsi **findKata()** yang menerima parameter berupa matrix dari puzzle dan sebuah string dari vector kata yang akan dicari. Fungsi ini mengembalikan **vector point yang berisi koordinat letak ditemukannya huruf dari kata** yang dicari yang nantinya digunakan untuk mendisplay. Langkah-langkah algoritma brute force adalah sebagai berikut:

#### 1. Mengassign:

- lenkata dengan panjang dari kata yang dicari
- idxMin untuk pencarian koordinat x dan y yaitu 0
- idxJMax yaitu untuk pencarian maksimal koordinat y batasnya sesuai banyaknya kolom pada matrix puzzle
- idxIMax yaitu untuk pencarian maksimal koordinat x batasnya sesuai banyaknya baris pada matrix puzzle

- Counter untuk menghitung banyaknya perbandingan yang telah dilakukan untuk mencari kata tersebut
2. Menggunakan ukuran baris dan kolom dari puzzle untuk looping pengecekan kata yang dicari
  3. Mengiterasi satu per satu character dalam puzzle dengan mengecek tempat dimana **letak huruf pertama** dari kata yang dicari
  4. Jika ditemukan tempat dari huruf pertama tersebut akan dilakukan **pengecekan apakah panjang dari kata pada titik tersebut memenuhi** untuk dicari ukurannya dalam matrix agar tidak terjadi index out of bounds. Pengecekan ini membutuhkan value idxMin, idxJMax, dan idxImax.
  5. Selanjutnya apabila panjang kata memenuhi ukuran matrix maka akan dilakukan **pengecekan dari huruf kedua sampai terakhir** dengan urutan atas, bawah, kiri, kanan, diagonal ke atas kiri, diagonal ke atas kanan, diagonal ke bawah kiri, diagonal ke bawah kanan (tentunya pengecekan dilakukan hanya jika memenuhi syarat batas ukuran matrix, jika tidak maka akan skip pengecekan).
  6. Prosedur pengecekan:
    1. Jika sudah memenuhi kriteria maka akan dilanjutkan dengan **membuat vector points yang kosong**
    2. Vector points ini berguna jika index ke-x dari sebuah string sama dengan elemen matrix[i][j] yang diiterate, maka koordinat i j akan **di push ke vector points**
    3. Apabila **banyak elemen dari vector points sama dengan panjang kata**, maka dipastikan koordinat-koordinat tersebut adalah jawabannya, jika tidak maka akan **dilanjutkan pengecekan** ke arah selanjutnya.
  7. Jika pengecekan tersebut benar dan ditemukan koordinat masing-masing huruf yang sesuai maka akan **di store ke suatu struct point**.
  8. Jika **tidak ditemukan**, akan mengeluarkan output **“Word doesn’t exist”**

## II. SOURCE PROGRAM

Struct Point, Function newMatrix dan newWord

```
1  #include <bits/stdc++.h>
2  #include <vector>
3  using namespace std;
4
5  struct Point
6  // struct ini berisi koordinat x dan y yang menampung letak huruf yang telah ditemukan
7  {
8      int x, y;
9  };
10
11 vector< vector<char> > newMatrix(const char* filename) {
12     // fungsi ini digunakan untuk membuat matrix sesuai input file user
13     vector< vector<char> > mat;
14     string line;
15     ifstream file;
16     file.open(filename);
17     if (file.is_open()) {
18         getline(file, line);
19         while(line != "") {
20             vector<char> temp;
21             for (int i = 0; i < line.length(); i+=2) {
22                 temp.push_back(line[i]);
23             }
24             mat.push_back(temp);
25             getline(file,line);
26         }
27         file.close();
28     } else {
29         cout << "failed to open the file" << endl;
30     }
31
32     return mat;
33 }
34
35 vector<string> newWord(const char* filename) {
36     // fungsi ini digunakan untuk membuat vector dari kata kata yang akan dicari
37     vector<string> words;
38     string line;
39     ifstream file;
40     file.open(filename);
41     if (file.is_open()) {
42         vector<char> temp;
43         getline(file, line);
44         while(line != "") {
45             getline(file,line);
46         }
47
48         while(getline(file,line)) {
49             words.push_back(line);
50         }
51         file.close();
52     } else {
53         cout << "failed to open the file" << endl;
54     }
55
56     return words;
57 }
```

## Procedure displayMatrix dan displayAnsMatrix

```
1 void displayMatrix(vector< vector<char> > mat){
2     // fungsi ini digunakan pada saat testing agar dapat mengecek inputan matrix yang dibaca dari file yang diinput
3     for (int i=0; i< mat.size(); i++){
4         for (int j=0; j< mat[i].size(); j++){
5             cout << mat[i][j] <<" ";
6         }
7         cout << "\n";
8     }
9 }
10
11 void displayAnsMatrix(vector< vector<char> > mat, vector<Point> p){
12     // fungsi ini digunakan untuk mendisplay matrix jawaban dari penyelesaian suatu kata
13     vector< vector<char> > mAns;
14     // membuat matrix berisikan "-"
15     for (int i=0; i< mat.size(); i++){
16         vector<char> temp;
17         for (int j=0; j< mat[i].size(); j++){
18             temp.push_back('-');
19         }
20         mAns.push_back(temp);
21     }
22     // memunculkan huruf yang ditemukan
23     for (int i=0; i<p.size();i++){
24         mAns[p[i].x][p[i].y] = mat[p[i].x][p[i].y];
25     }
26
27     displayMatrix(mAns);
28 }
29
```

## Function FindKata

```
1  vector<Point> findKata(vector< vector<char> > m,string kata, int* totalComp){
2      // fungsi ini digunakan untuk melakukan pencarian koordinat letak suatu kata ditemukan
3      int lenKata = kata.length();
4      int idxMin = 0;
5      int idxJMax = m[0].size();
6      int idxIMax = m.size();
7      int counter = 0;
8      for (int i=0; i<m.size(); i++){
9          for (int j=0; j<m[0].size();j++){
10             counter++;
11             // mengecek apakah elemen pertama dari kata yang dicari sesuai dengan elemen matrix
12             if (m[i][j] == kata[0]){
13                 // mengecek lurus ke atas
14                 if (i-(lenKata-1)>=idxMin){
15                     vector<Point> Points;
16                     for(int x = 0; x<lenKata; x++) {
17                         if (m[i-x][j] != kata[x]) {
18                             break;
19                         } else {
20                             Point P;
21                             P.x = i-x;
22                             P.y = j;
23                             Points.push_back(P);
24                         }
25                     }
26                     if (Points.size() == lenKata) {
27                         *totalComp += counter;
28                         cout <<"Number of Comparisons: "<<counter<<"\n";
29                         return Points;
30                     }
31                 }
32                 // mengecek lurus ke bawah
33                 if (i+(lenKata-1)<=idxIMax){
34                     vector<Point> Points;
35                     for(int x = 0; x<lenKata; x++) {
36                         if (m[i+x][j] != kata[x]) {
37                             break;
38                         } else {
39                             Point P;
40                             P.x = i+x;
41                             P.y = j;
42                             Points.push_back(P);
43                         }
44                     }
45                     if (Points.size() == lenKata) {
46                         *totalComp += counter;
47                         cout <<"Number of Comparisons: "<<counter<<"\n";
48                         return Points;
49                     }
50                 }
51             }
52         }
53     }
```

```

1  // mengecek lurus ke kiri
2  if (j-(lenKata-1)>=idxMin){
3      vector<Point> Points;
4      for(int x = 0; x<lenKata; x++) {
5          if (m[i][j-x] != kata[x]) {
6              break;
7          } else {
8              Point P;
9              P.x = i;
10             P.y = j-x;
11             Points.push_back(P);
12         }
13     }
14     if (Points.size() == lenKata) {
15         *totalComp += counter;
16         cout <<"Number of Comparisons: "<<counter<<"\n";
17         return Points;
18     }
19 }
20 // mengecek lurus ke kanan
21 if (j+(lenKata-1)<=idxJMax){
22     vector<Point> Points;
23     for(int x = 0; x<lenKata; x++) {
24         if (m[i][j+x] != kata[x]) {
25             break;
26         } else {
27             Point P;
28             P.x = i;
29             P.y = j+x;
30             Points.push_back(P);
31         }
32     }
33     if (Points.size() == lenKata) {
34         *totalComp += counter;
35         cout <<"Number of Comparisons: "<<counter<<"\n";
36         return Points;
37     }
38 }
39 // mengecek diagonal ke arah atas kiri
40 if (j-(lenKata-1)>=idxMin && i-(lenKata-1)>=idxMin){
41     vector<Point> Points;
42     for(int x = 0; x<lenKata; x++) {
43         if (m[i-x][j-x] != kata[x]) {
44             break;
45         } else {
46             Point P;
47             P.x = i-x;
48             P.y = j-x;
49             Points.push_back(P);
50         }
51     }
52     if (Points.size() == lenKata) {
53         *totalComp += counter;
54         cout <<"Number of Comparisons: "<<counter<<"\n";
55         return Points;
56     }
57 }

```

```

1 // mengecek diagonal ke arah atas kanan
2 if (j+(lenKata-1)<=idxJMax && i-(lenKata-1)>=idxMin){
3     vector<Point> Points;
4     for(int x = 0; x<lenKata; x++) {
5         if (m[i-x][j+x] != kata[x]) {
6             break;
7         } else {
8             Point P;
9             P.x = i-x;
10            P.y = j+x;
11            Points.push_back(P);
12        }
13    }
14    if (Points.size() == lenKata) {
15        *totalComp += counter;
16        cout <<"Number of Comparisons: "<<counter<<"\n";
17        return Points;
18    }
19 }
20 // mengecek diagonal ke arah bawah kiri
21 if (j-(lenKata-1)>=idxMin && i+(lenKata-1)<=idxIMax){
22     vector<Point> Points;
23     for(int x = 0; x<lenKata; x++) {
24         if (m[i+x][j-x] != kata[x]) {
25             break;
26         } else {
27             Point P;
28             P.x = i+x;
29             P.y = j-x;
30             Points.push_back(P);
31         }
32     }
33     if (Points.size() == lenKata) {
34         *totalComp += counter;
35         cout <<"Number of Comparisons: "<<counter<<"\n";
36         return Points;
37     }
38 }
39 // mengecek diagonal ke arah bawah kanan
40 if (j+(lenKata-1)<=idxJMax && i+(lenKata-1)<=idxIMax){
41     vector<Point> Points;
42     for(int x = 0; x<lenKata; x++) {
43         if (m[i+x][j+x] != kata[x]) {
44             break;
45         } else {
46             Point P;
47             P.x = i+x;
48             P.y = j+x;
49             Points.push_back(P);
50         }
51     }
52     if (Points.size() == lenKata) {
53         *totalComp += counter;
54         cout <<"Number of Comparisons: "<<counter<<"\n";
55         return Points;
56     }
57 }
58 }
59 }
60 }
61 vector<Point> Points;
62 return Points;
63 }

```



## Function Main

### III. SCREENSHOT INPUT DAN OUTPUT

\*note: untuk keterbacaan yang lebih baik, pembacaan lampiran dapat dizoom, grid pada lampiran dilakukan agar lebih rapi

Awal Program Berjalan:

```
 8 8 8      8eeee8
8 8 8 eeeee eeeee eeeee 8      eeee eeee eeee e e e
8e 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
88 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
88 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
88ee8ee8 8eee8 88 8 88ee8 8eee88 88ee 88 8 88 8 88e8 88 8

File Name Input (please include .txt when input) : 
```

Small case (3)

Small Case - 1 ( 14 x 12, 17 words) → 0,013 s

Input: small1.txt

```
test > small1.txt
1  G R P E R S I M M O N Q
2  B I E X F F I A P O L K
3  Z M A U L X M J B E N I
4  X X R G R A P E M O W M
5  H A K V L I F O L I A E
6  C P W C I V N E K P P L
7  A P C E M L M N O O R O
8  E L N A E R B I R K I N
9  P E P N E A S R A A C N
10 W A U T N N O E N X O S
11 V R A A F I G G G M T Z
12 P W N O U K X N E C P Y
13 U A Z M U L P A D E D M
14 D G Z X I W N T P C H F
15
16 APPLE
17 APRICOT
18 BANANA
19 FIG
20 GRAPE
21 KIWI
22 LEMON
23 LIME
24 MELON
25 ORANGE
26 PEACH
27 PEAR
28 PERSIMMON
29 PLUM
30 PRUNE
31 TANGERINE
32 WATERMELON
```

Total comparison: 1324

[illegible][illegible]

Input: small2.txt

## Output

[illegible]



Input: small3.txt

## Output

<p>1. ANNA Number of Comparisons: 155</p> <p>2. ARIEL Number of Comparisons: 81</p> <p>3. AURORA Number of Comparisons: 250</p>	<p>2. ARIEL Number of Comparisons: 81</p> <p>4. BELLE Number of Comparisons: 2</p> <p>5. JASMINE Number of Comparisons: 83</p>	<p>5. JASMINE Number of Comparisons: 83</p> <p>6. MERIDA Number of Comparisons: 73</p> <p>7. MULAN Number of Comparisons: 55</p>	<p>7. MULAN Number of Comparisons: 55</p> <p>8. POCAHONTAS Number of Comparisons: 72</p> <p>9. RAPUNZEL Number of Comparisons: 116</p>	<p>9. RAPUNZEL Number of Comparisons: 116</p> <p>10. SNOWWHITE Number of Comparisons: 38</p> <p>Time taken : 0.005900s Total comparison: 917</p>
---	--	--	--	--

### Medium case (3)

Medium Case - 1 ( 20 x 18, 26 words) → 0,024 s

Input: medium1.txt

```
test > ≡ medium1.txt
1  F L W O J D S E P D V C H I C K E N
2  G L I A N S Q J C K E S O I Y H A D
3  E I O W O W X Z U C W E G N L K O C
4  A Z C W B F K S I A L Y R U K H Y W
5  X A O D U S R M L K Y B Q Z C A L L
6  Z R W G I V L L V D E B R B L S Q I
7  B D M K F U O N Z A V E F E R F B M
8  Z J O L T W Q S Q O T B E D R E I W
9  B T Y U H K U S N T W O R R A P S G
10 Q N X M B R J U O W A U K Y E K M Z
11 T O U T L T J P H X T R A M N B H R
12 A F J A S I X O Z P M D N L B B N S
13 J Q W D T O R T O I S E G F F J W C
14 H S Y S R R C C F H G T A C X E K H
15 N R T C J E A O F C B I R D K L N E
16 M L N B R D T T G V F T O L U G S E
17 X A Z Q N I J S S U P X O Y Z G K T
18 U E Y A U P Z W M G V T W T L U O A
19 Q S P U F S I K D A I Z I R V N Q H
20 E E Z N A P M I H C H P I C Q P M S
21
22  B I R D
23  C A T
24  C H E E T A H
25  C H I C K E N
26  C H I M P A N Z E E
27  C O W
28  D E E R
29  F L Y
30  H A M S T E R
31  K A N G A R O O
32  L I Z A R D
33  O C T O P U S
34  O T T E R
35  O W L
36  P A N D A
37  P I G
38  R A T
39  S E A L
40  S N A I L
41  S P A R R O W
42  S P I D E R
43  S Q U I D
44  S W A L L O W
45  T O R T O I S E
46  W A L R U S
47  W O L F
```

Total comparison: 4659



CONTENTS

W  
A  
L  
R  
U  
S

```
Time taken : 0.024000s
Total comparison: 4659
```

Medium Case - 2 ( 24 x 22, 18 words) →0,012 s

Input: medium2.txt

```
test > ≡ medium2.txt
1 | A H O I Y C M P I T N U P M L L X O M Y P E
2 | B N N R C R P W K C T J H G P U Z D X F S G
3 | K Z Y T T E R C O O K I E S A N D C R E A M
4 | M P Y R W Z R E U B J F R P O W D J I R L X
5 | T B B Q R T P Q B F L Z A X O Z R D M N Q S
6 | F X B E M E P L R P B U T T E R P E C A N H
7 | Z E T R W E H Y B R S A E Z W E F G K O N P
8 | P G S B X Z I C C D F A Q B K D S Y P V P Y
9 | D O P R F E C P S X Q W R A E B K H M G P Q
10 | Q R X U O D P E A C H R L W R R T L C G N A
11 | X O D V C O C O N U T L V O S U R O F Z S K
12 | E G F C U X P R L G I K K X N U G Y L T B U
13 | E U C F F N N R N N P P N L O N F O R N I J
14 | F V U M U X K L A R Q H A S A F W A V W R G
15 | F R O S C A F V T L V W P M Q A W G Z R N Q
16 | O F R A Z F H K I N I O X F I B G Y S P D S
17 | C K M Y F C U J L C D N Q C E T A Q K B D L
18 | L L K U N J F N O M N Z E R P E M U S W J R
19 | J P S E S E E R P F H G R P U Z I G S T L Q
20 | K U R L M L A O A B A Y D H E Y C J L R Q S
21 | F F V P Y W H Q E N Q I S Y H C V D T B J Y
22 | Q E N R K M C R N T F T Y N R I A G M P Y U
23 | G F Z P K G O Z X D B T J M K T Z N N I L V
24 | B T P R V E M O L I D E U X C A N A N A B V
25 |
26 | BANANA
27 | BLUEBERRY
28 | BUTTERPECAN
29 | CHERRY
30 | COCONUT
31 | COFFEE
32 | COOKIESANDCREAM
33 | FRENCHVANILLA
34 | FUDGE
35 | MANGO
36 | MOCHA
37 | NEAPOLITAN
38 | PEACH
39 | PRALINEPECAN
40 | RASPBERRY
41 | REESES
42 | STRAWBERRY
43 | WALNUT
```

Output

Total comparison: 5053

<div>1. BANANA Number of Comparisons: 527</div> <div>ANANAB</div>	<div>3. BUTTERPECAN Number of Comparisons: 121</div> <div>BUTTERPECAN</div>	<div>5. COCONUT Number of Comparisons: 225</div> <div>COCONUT</div>	<div>7. COOKIESANDCREAM Number of Comparisons: 52</div> <div>COOKIESANDCREAM</div>	<div>9. FUDGE Number of Comparisons: 181</div> <div>FUDGE</div>
<div>2. BLUEBERRY Number of Comparisons: 76</div> <div>BLUEBERRY</div>	<div>4. CHERRY Number of Comparisons: 162</div> <div>CHERRY</div>	<div>6. COFFEE Number of Comparisons: 353</div> <div>COFFEE</div>	<div>8. FRENCHVANILLA Number of Comparisons: 442</div> <div>FRENCHVANILLA</div>	<div>10. MANGO Number of Comparisons: 332</div> <div>MANGO</div>
<div>11. MOCHA Number of Comparisons: 513</div> <div>MOCHA</div>	<div>13. PEACH Number of Comparisons: 285</div> <div>PEACH</div>	<div>15. RASPBERRY Number of Comparisons: 189</div> <div>RASPBERRY</div>	<div>17. STRAWBERRY Number of Comparisons: 241</div> <div>STRAWBERRY</div>	
<div>12. NEAPOLITAN Number of Comparisons: 471</div> <div>NEAPOLITAN</div>	<div>14. PRALINEPECAN Number of Comparisons: 249</div> <div>PRALINEPECAN</div>	<div>16. REESES Number of Comparisons: 484</div> <div>REESES</div>	<div>18. WALNUT Number of Comparisons: 328</div> <div>WALNUT</div>	
<div>13. PEACH Number of Comparisons: 285</div> <div>PEACH</div>	<div>15. RASPBERRY Number of Comparisons: 189</div> <div>RASPBERRY</div>	<div>17. STRAWBERRY Number of Comparisons: 241</div> <div>STRAWBERRY</div>	<div>Time taken : 0.012000s Total comparison: 5053</div>	

Medium Case - 3 ( 22 x 22, 18 words) → 0,01 s

Input : medium3.txt

```
test > ≡ medium3.txt
1  E E L A D E R I A C K A T F H Q J D I N K S
2  K A S P I J P S A K T X S S W K R R Q O E I
3  K Z O P O H B O R D E R C O L L I E W Q G X
4  O S X O P M W J C Z F H M C R B F S I Q N R
5  E T X I Q S E C W B U L L D O G J K P J K I
6  L E I N A P S R E K C O C S J D T F S M A V
7  D O L T C M J E A Z Y P Q G M V D D Q X D T
8  U N O E K D Y X Q N Q P E I M Z X N Q B Y B
9  L B U R A G J O W C I N A C S L I I B L U Y
10 M X F H Z X Q B J Q S A G C M G O L Q O D B
11 C I L Q S G W Y P S I I N Z A M R S F O A N
12 I R C B B H K L M H Z T I P S D P C Y D D A
13 Q I D K A M C Y W E L A X O J Y X T T H L M
14 P S V T L O Z A V E C M A U C Q X J H O Y R
15 Y H S E W S V J D P Y L U P Q W Y W L U Z E
16 U S Q Z W P V H E D I A Z A Q B D S T N E B
17 Q E D X Q A I G R O C D T Y Q E A D M D T O
18 K T E D P A A B C G N B H B T W X S L H R D
19 O T C U X S U M J O T K I W L Z S T E O F L
20 N E L L Z A J O L Y W R H E H Z C X Y N C Q
21 J R I N G H D M B M T C S R P R Y W I X J T
22 R R N T B L T T X A P S U V R G R M Q Y F I
23
24 AIREDALE
25 BASENJI
26 BLOODHOUND
27 BORDERCOLLIE
28 BOXER
29 BULLDOG
30 COCKERSPANIEL
31 CORGI
32 DACHSHUND
33 DALMATIAN
34 DOBERMAN
35 IRISHSETTER
36 LHASAAPSO
37 POINTER
38 POMERANIAN
39 SHEEPDOG
40 SHIHTZU
41 SPITZ
```

# Output

## Total comparison: 4151

<div>1. AETHIOPE Number of Comparisons: 9</div> <div>E L A D E R I A</div>	<div>3. BLOODHOUND Number of Comparisons: 174</div> <div>B L O O D H O U N D</div>	<div>5. BOMBER Number of Comparisons: 206</div> <div>B O M B E R</div>	<div>6. BULLDOG Number of Comparisons: 98</div> <div>B U L L D O G</div>	<div>8. C2001 Number of Comparisons: 363</div> <div>C 2 0 0 1</div>
<div>2. BAGEROT Number of Comparisons: 346</div> <div>B A G E R O T</div>	<div>4. BORDERCOLLIE Number of Comparisons: 51</div> <div>B O R D E R C O L L I E</div>	<div>6. BULLDOG Number of Comparisons: 98</div> <div>B U L L D O G</div>	<div>7. COCKERSPANIEL Number of Comparisons: 123</div> <div>C O C K E R S P A N I E L</div>	<div>9. DACHSHUND Number of Comparisons: 317</div> <div>D A C H S H U N D</div>
<div>10. GALMATION Number of Comparisons: 364</div> <div>G A L M A T I O N</div>	<div>12. IREISHGETTER Number of Comparisons: 222</div> <div>I R E I S H G E T T E R</div>	<div>14. POINTER Number of Comparisons: 48</div> <div>P O I N T E R</div>	<div>15. POMERANIAN Number of Comparisons: 26</div> <div>P O M E R A N I A N</div>	<div>17. SHIHITU Number of Comparisons: 453</div> <div>S H I H I T U</div>
<div>11. COCKERHORN Number of Comparisons: 396</div> <div>C O C K E R H O R N</div>	<div>13. SHAGBARK Number of Comparisons: 468</div> <div>S H A G B A R K</div>	<div>15. POMERANIAN Number of Comparisons: 26</div> <div>P O M E R A N I A N</div>	<div>16. SHEEPDOG Number of Comparisons: 230</div> <div>S H E E P D O G</div>	<div>18. SPITZ Number of Comparisons: 257</div> <div>S P I T Z</div>
<div>12. IREISHGETTER Number of Comparisons: 222</div> <div>I R E I S H G E T T E R</div>	<div>14. POINTER Number of Comparisons: 48</div> <div>P O I N T E R</div>	<div>16. SHEEPDOG Number of Comparisons: 230</div> <div>S H E E P D O G</div>	<div>17. SHIHITU Number of Comparisons: 453</div> <div>S H I H I T U</div>	<div>Time taken : 0.030000s Total comparisons: 4151</div>

### Large case (3)

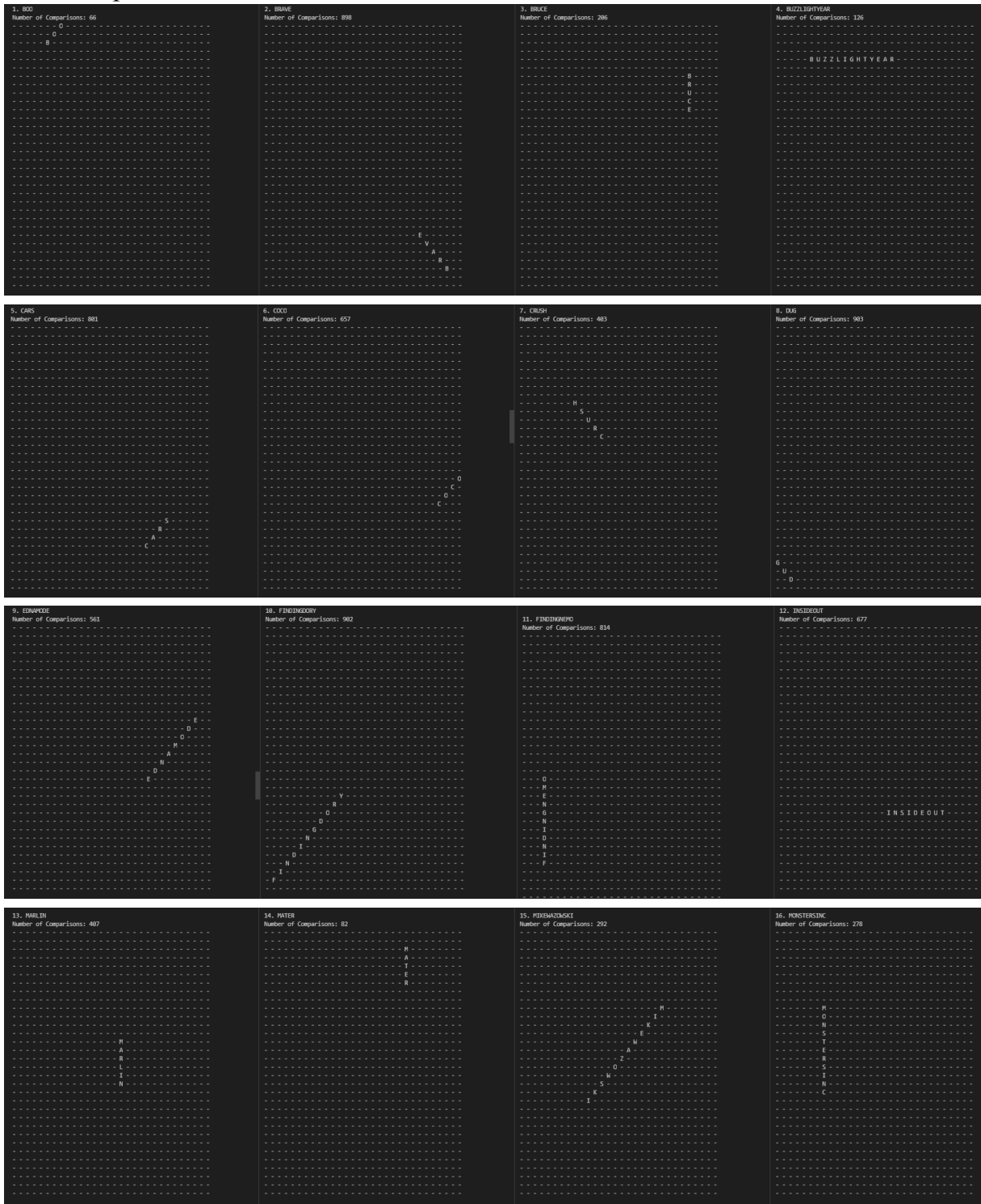
Large Case - 1 ( 32 x 30, 28 words) → 0,027s

Input: largel1.txt

```
test > E largel1.txt
1 D K M A Q Q N O T D Q L D K D O Z X F M M N E Q R D A S I P
2 T B W M A G O W I P T T U N I Z F F J G D C Y Y Y E D I H S
3 K N G J O B T T V R K R A E H M Z U R D L M K R B W M L B Y
4 I Z M W N C O Y Q X H V P Q Z U P Y A U F A W O L N D Y N E
5 M F C R A B U Z Z L I G H T Y E A R G O X T N N M Y G H D F
6 Q D H Q Y R A S N L F P E M E Q Y D W O I E H F W X N S L O
7 U B P Q D W W T L X Z H Z C K I G R G M Z R E T B B T D H Q
8 W J X F O A W U W O L I R J G W P L O H O M X A H R V Z R B
9 M V J G O U S L T O W I H O S M D Y J T C Z S D J U R K L B
10 C E U O W M M M H C O J B Z H C C H H C S M D C R C M M I Y
11 O L F D F S M O F S N B N N X G B J G U I Y D E C E H O S B
12 S T P Y F Q R N G K U R X S A O V X A K H Y O Z G L G E X U
13 B G W K I G H S U T C R W E D P Y N E C T V N T G M D H D C
14 O P P N R T R T I H J D C V Y Q M W O L Q O Q Q G O R S P R
15 S S V I E X D E S E D L U D U E A P Q V O P P E M R E X Y A
16 K Z T L H H R R Z G C W N D F Z R B U O R Q D A Y U C N T T
17 Q D S S N B S K O X I Y D O S L A N H U J N C I K H H N A
18 W X S O R E A I B O Z B D W B Q I Q K E F D E R G O E J P T
19 I F F M K B F N C D P S S W G Z N O A A E P G K X I B D E O
20 C Z U E B J I C N D L K A A R Z V G Z E N H E Z N W P P C U
21 C W G N U B V R P I I Y G M R X D T U M W Z Y C R H A O W I
22 D R H G N Q Y Y B N R H P L L Q T Z Q X N U R R V G C L M L
23 I J Q N N A Z X S O S W U N J V I N S I D E O U T F U F L L
24 E E M I V O A I D S H N U X H E Y C W V D A S S H F H X R E
25 V T W D H Y L G Y A A X Z L H Z W P S I H I R S W N F I O V
26 F C R N Q G N P K U V O O L M Q O V B G R A W E I X J O C O
27 X A Z I U I E Z V R Q Z X L L A P L O S C N E L V V D V N E
28 Q N N F D G R Z Z Z T K Q J E K E H I R H X H L W A N R V M
29 G T E N E H J H W U R K I V I S R Z M G N C J X K O R G G P
30 C U I E R R A D L A W Z T S N B S O E C D I U F M J H B N Q
31 G F D V U V Z T F O P J K Y H S Z Z R M C R O Q A H L V X L
32 W R I T O B L K B T W Z S S B S F K E Z F T Z O A N A L A G
33
34 BOO
35 BRAVE
36 BRUCE
37 BUZZLIGHTYEAR
38 CARS
39 COCO
40 CRUSH
41 DUG
42 EDNAME
43 FINDINGDORY
44 FINDINGNEMO
45 INSIDEOUT
46 MARLIN
47 MATER
48 MIKEWAZOWSKI
49 MONSTERSINC
50 RATATOUILLE
51 REMY
52 REX
53 RUSSELL
54 SHERIFFWOODY
55 SLINKYDOG
56 SULLIVAN
57 THEGOODDINOSAUR
58 THEINCREDIBLES
59 TOYSTORY
60 UP
61 WALLE
```

# Output

## Total comparison: 12741



16. *Survey*

21. SHEET#FAZEY  
Number of Comparisons: 485



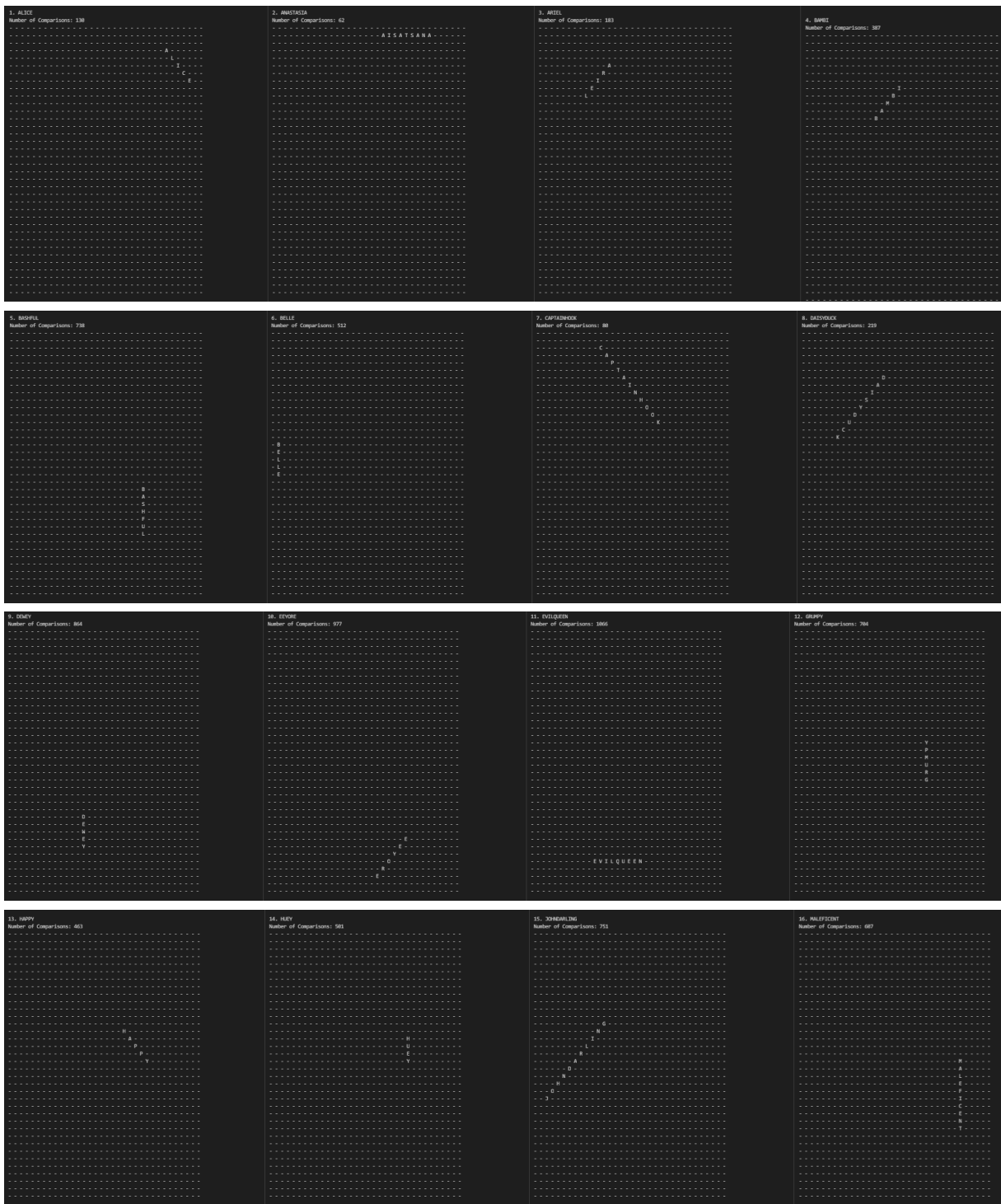
Large Case - 2 ( 36 x 34, 31 words ) →0,31s

Input : large2.txt

```
test > large2.txt
1  R A Y F S U F Z P J J D F S H X P F I A W T P G M V W E U R X Q P L
2  J U R X A H C P I O O P A B L R P G F A I S A T S A N A I C H F D V
3  Q A T X C V X J D O X C D S C Q E E W A K P Q W J S R V X K S W S F
4  A I T N D H W X H M M K A S A Y Y V C L Y S H Z H Y X A S D D D L E
5  C R F T B D B N S L N Z J P W B U V H S Q N O K E X N N L H D Q O P
6  T Q R F K U I Q E E Y J A J T H N I C J K L B A Y N P O V I E J W G
7  A K I T P B G X I B E R S H D A S W O U H C Y C C K T A M N C O H D
8  H E L M O D G O O I I W Y A K Q I I L B D P R S O O Y E W G O E Y P
9  Y B O R U W A M L E A Y I R I B Q N I H X C Y C K F A K J J N Z G D
10 Z B O R S F B U L O I S I O M I N N H A P Z M W L Y T Y L L B U L B
11 S A T N O H A C O P Y Q U A R Q X I T O O E H E U J E N D Q J A D T
12 Z F Y R H L S A T D R U B E C L U E X L O S P U G I W W A N W O P V
13 M I C K E Y M O U S E R G O O Z A T Z T K K D G F O K J H Y X I J J
14 B O A K T F K C R D U N K X Y H F H U I H Y B X M I O A R L W T L Q
15 D T O A G O K D F D I V R O I E H E N W K A P H H S I R D U Z S A R
16 B B G E Z T P N A L P W R Z Z X I P C A W P P Y U L C M C F A B F M
17 N E G B U C B H R A T J J T Y F X O Z M H J H P E D O J F S T D A P
18 G L X G B I U A V Y Z I G B P L K O Z G N K B M Y U S Q M L V L F L
19 E L P C R X D J M M L P H I F R F H A G R M E U I R M U A N P W V U
20 U E D Z B N L R I G E R Z L G I Y J E P G X A R J F Y A L J H L Z T
21 Y Q Q K H X I E W O S F B G U R Q J E W W P S G E U B M E Q N N K O
22 P A R O T F F O S E L W N C Z E A A Q G N X A B B H C G F C P X U K
23 E J J E H D M H T Z B W E E G Z O A P W X K F A K H S Z I L Y I F W
24 N K F K U Y R O M S J M L I A W A H C G S G U S Q A Q P C Y Q L I B
25 L O F W P O W F X I V L Y A H U W M G H V Z M H G O P H E P P U E C
26 C N C T B Q W M Y K N O O D O B Y L J K U P Y F L N B F N E G O D O
27 L A Q A J X O M X G J N A E R W W K H W Y Y G U N E R A T E S I X M
28 H M W L G L R C L A O C I W R Z J N P R Z U T L P F N B H L O D R Y
29 O Y P Y G V N I I Z C L M E Z V A Q K F D T X Y E O Q W M S R C Z F
30 N J Q D I M G U S P H Q U Y M S N O W W H I T E H K F O F S P S V R
31 Z Y M D X U L D A X J M N D U O B B B I Z O Y A F R K T K P N F W E
32 G U S R C G A W S G R E V I L Q U E E N K O V L Q U T F Z L G B W R
33 E B K O T T M O O F O G W F X H H S P L R S E F R C S T G E S X I O
34 B C I Z N Y M C H H Y W S Z J A E V E E W F U X Q H T F A K S G C E
35 D P U J O M U L L H F S J J Q D P A E U Q U E E N O F H E A R T S X
36 C F N V R T T U Y D N E W X H Z X O G C L C Z F W L O J H W P N E A
37
38 ALICE
39 ANASTASIA
40 ARIEL
41 BAMBI
42 BASHFUL
43 BELLE
44 CAPTAIN HOOK
45 DAISY DUCK
46 DEWEY
47 EEYORE
48 EVIL QUEEN
49 GRUMPY
50 HAPPY
51 HUEY
52 JOHN DARLING
53 MALEFICENT
54 MICKEY MOUSE
55 MINNIE MOUSE
56 MOWGLI
57 MUFASA
58 PLUTO
59 POCAHONTAS
60 QUEEN OF HEARTS
61 ROBIN HOOD
62 SCROOGE MCDUC
63 K
64 SHEREKHAN
65 SLEEPY
66 SNOW WHITE
67 WENDY
68 WINNIE THE POOH
```

# Output

Total comparison: 18266



17. MICKEYMOUSE  
Number of Comparisons: 489

MICKEYMOUSE

18. MINNIEPOOH  
Number of Comparisons: 791

19. MIMM1  
Number of Comparisons: 795

20. MIFASA  
Number of Comparisons: 839

21. PLUTO  
Number of Comparisons: 578

22. PICAKENTAS  
Number of Comparisons: 398

SATNOHACOP

23. QUEENOFHEARTS  
Number of Comparisons: 1177

QUEENOFHEARTS

24. ROBINHOOD  
Number of Comparisons: 326

25. SCROOGEDUCK  
Number of Comparisons: 524

26. E  
Number of Comparisons: 89

27. SHERIDAN  
Number of Comparisons: 779

28. SLEEPY  
Number of Comparisons: 982

29. SNOWWHITE  
Number of Comparisons: 1882

SNOWWHITE

30. MENDY  
Number of Comparisons: 1283

YDNEW

31. WINNIETHEPOOH  
Number of Comparisons: 222

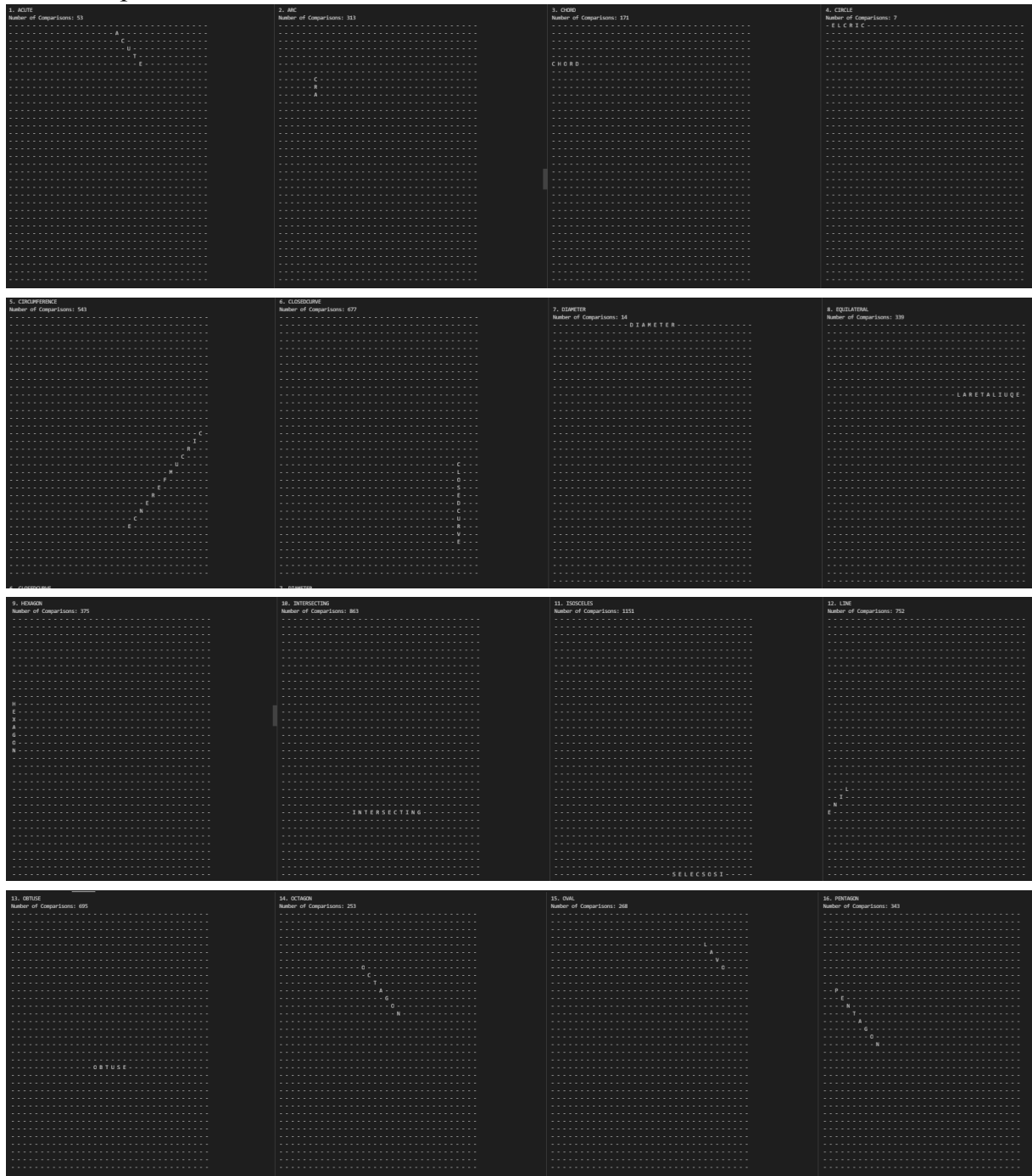
Time taken : 0.031080s  
Total comparison: 18266  
PS D:\IP\Semester4\atma\tuc11\github\tuc11stima-3\srcs

Large Case - 3 ( 34 x 34, 29 words ) →0,029s

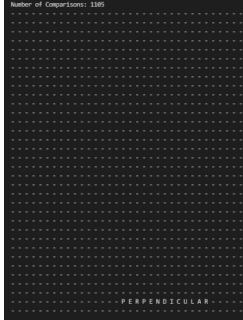
Input: large3.txt

```
test > large3.txt
1  M E L C R I C U P Z Z F L D I A M E T E R E Y U Q W S R T U W S T Z
2  F Q O I N T F O P L W G K U H M W D A K S H J M V K M J D J P G G L
3  X A X K Y W M B O D K G B W C H P P C C L L C S H C T Z M P J A Z V
4  D I O V Y Q T X S U J Y R P X H M Y Q U U G E U B E J S K J Z K J L
5  B I X E L B X F T B F C P T P Z H C I V K T D F G T L L B F F U M O
6  C H O R D U G Z I R N K V R Z E V U E T K V E V J X M A K A L R H V
7  U N X Z E B A W F A J K J S Y D R X Q T T S E M Y I K K V W F F F C
8  V L V U E R C I I C G A P A O Q I C P J H Z S R T Y O O M O K I R I
9  J Y R G B P R K L X X P W R U C Q N W X E Z J P T J Z Z K Q F G L H
10 U W A D P Z A I F V C I P A W E T L W Z W R L A R E T A L I U Q E I
11 M S P L E F C R Q C H F A J W G C A T K T A W M W Z X X D H E B Q I
12 H L M E W N S Y T J U W E Q J V V R G G S D S A K E E T Y P P X D N
13 E X N G N R L A I Z S Z F N W W J E U O R I G H T A N G L E W H P S
14 X L G X J T M Z H L R I F B H S K T B V N U F X B P T E W Q C P S X
15 A N G Y K W A S X N V G E B S I D A P K N S K P E C I E L Q I L A Y
16 G L U N K J J G P A J L Z V E W G L J P O A V N R R B U Y A N Q C L
17 O H C E A V G B O I G J L C Y D P I B V Y N Y S I I A K M M C I N F
18 N J V N K I M X S N R H T X B T C R W L S F Y W H F L U W G R S D E
19 X D F J R N R W A H T Q L O O W W D G Y E M G W K Z N C Q C V Y A L
20 Q H V S Y R J T L A H Y Q B A B M A Z X A G F S T O A Z U S C G D X
21 L V Z N X Z C E S G H C N A O B T U S E U H R T B C O M P N L G K H
22 G U P C R E O U Y E K Y T N N W L Q T Q D L P G M V F R L R O L M B
23 A P B L R W Z N I P G O Y Q N E W O Y Z B L I J E E O B B G S E V P
24 E Y I N J H F T I U Q M E T P A K R X V X M F L R D N U X S E U Q G
25 K N I J V H W K A H S G E P T K T L N P T A M E C Z I A C B D U F A
26 E O R H O M B U S O L O I N T E R S E C T I N G R D G S E J C S Y H
27 M L E V E J C X K E P O I N T V K G T D V C B Y K H Y U J F U O E E
28 C K L Z U C M N F I J W L S O A E S C E E E T J E S T L Q H R R F P
29 M Y D A F O K F S K V W F X M H M A P V A H N Q J C B I V S V L T S
30 M B H M Z D V Y L Y A W U E N O C Y M B A J X L F O H X F J E M B J
31 X B E B G I N S R Q R Q X I Z S G I E R R H Y K F O Q P W O I D W T
32 C P K A D V B V K C C B S R P R D G Z P R J O E V E K L F T K P F C
33 S O U I C N W I Q D R I N S S P E R P E N D I C U L A R N W U L Y
34 H D F N H K E G V T W I M H I Q F T D V S E L E C S O S I F I S O V
35
36 ACUTE
37 ARC
38 CHORD
39 CIRCLE
40 CIRCUMFERENCE
41 CLOSED CURVE
42 DIAMETER
43 EQUILATERAL
44 HEXAGON
45 INTERSECTING
46 ISOSCELES
47 LINE
48 OBTUSE
49 OCTAGON
50 OVAL
51 PENTAGON
52 PERPENDICULAR
53 POINT
54 QUADRILATERAL
55 RADIUS
56 RECTANGLE
57 RHOMBUS
58 RIGHTANGLE
59 SCALENE
60 SEGMENT
61 SQUARE
62 TRAPEZOID
63 TRIANGLE
64 VERTEX
```


Output  
Total comparison: 15116




17. PERPENDICULAR  
Number of Comparisons: 185



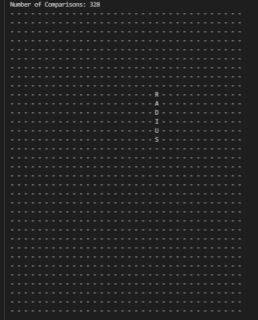
18. POINT  
Number of Comparisons: 89



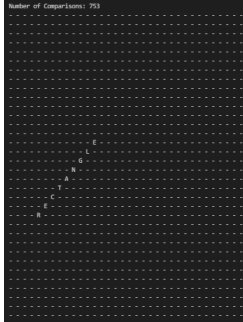
19. QUADRILATERAL  
Number of Comparisons: 732




20. RADIUS  
Number of Comparisons: 328




21. RECTANGLE  
Number of Comparisons: 793



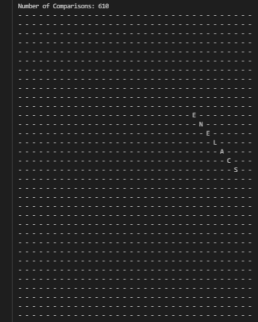
22. RHOMBUS  
Number of Comparisons: 853



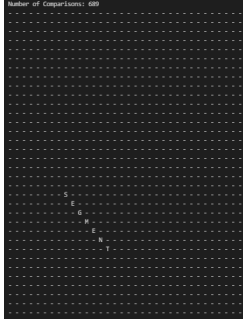
23. RIGHTANGLE  
Number of Comparisons: 429




24. SQUARE  
Number of Comparisons: 638




25. SEAGULL  
Number of Comparisons: 689



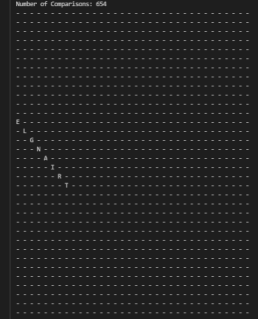
26. SQUAD  
Number of Comparisons: 676



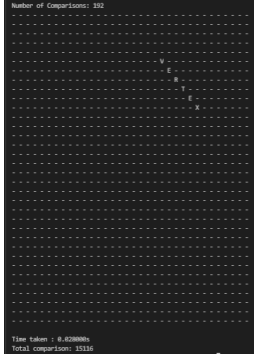
27. THIRTIETH  
Number of Comparisons: 383



28. TRIANGLE  
Number of Comparisons: 656



29. VERTEX  
Number of Comparisons: 351



Time taken : 8.62888ms  
Total comparison: 3333

Word not found case (1) → mengeluarkan “Word doesn’t exist”

Word not found case -1 ( 16 x 16, 5 words) → 0,004s

\*note: there is no “effort” in puzzle

Input: test1.txt

## Output

Total comparison: 497

[illegible]

#### IV. ALAMAT GITHUB KODE PROGRAM

<https://github.com/graceclaudia19/tucilStima-1.git>

## V. CHECK LIST

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan (no syntax error)	✓	
2. Program berhasil running	✓	
3. Program dapat membaca file masukan dan menghasilkan keluaran	✓	
4. Program berhasil menemukan semua kata di dalam puzzle	✓	