TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA

PENYELESAIAN WORD SEARCH PUZZLE DENGAN ALGORITMA BRUTE FORCE

Disusun oleh

Christine Hutabarat (13520005)

TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG 2022

DAFTAR ISI

I. ALGORITMA BRUTE FORCE

Algoritma *brute force* adalah salah satu algoritma yang langsung, sederhana, dan didasarkan pada definisi dan konsep pada persoalan yang akan diselesaikan. Algoritma *brute* force dapat menyelesaikan hampir semua persoalan. Salah satu persoalan yang dapat diselesaikan oleh algoritma ini adalah permainan mencari kata atau *word search puzzle.* Permainan ini mengharuskan pemainnya untuk mencari kata-kata yang telah diberikan pada suatu segiempat yang disusun oleh huruf-huruf. Kata-kata tersebut dapat tersusun ke delapan arah yang berbeda, yaitu ke atas, bawah, kanan, kiri, kanan atas, kanan bawah, kiri atas, dan kiri bawah.

Algoritma *brute force* yang diterapkan pada permainan ini akan meninjau satu per satu huruf pada segiempat, kemudian membandingkan huruf-huruf yang mengikutinya di delapan arah yang berbeda dengan kata yang sedang dicari. Jika kata ditemukan, maka algoritma akan berhenti dan mulai mencari kata baru. Sementara itu, jika kata tidak ditemukan, akan ditinjau huruf pada posisi lainnya di segiempat yang belum ditinjau. Adapun perbandingan antara kata yang sedang dicari dengan huruf-huruf dalam segiempat pada suatu arah dilakukan dengan mensejajarkan kata yang sedang dicari dengan rentetan huruf dalam segiempat pada arah tertentnu. Jika urutan dari huruf-huruf pada kata yang dicari dan pada segiempat sama, maka kata dinyatakan telah ditemukan, dan akan dicari kata berikutnya. Namun, jika kedua huruf tersebut berbeda, maka akan dilakukan perbandingan ke arah lain.

II. SOURCE PROGRAM

Program disusun dalam tiga bagian, yaitu dua bagian struktur data abstrak berupa matriks dan list berkait, serta bagian kode untuk program utama. Keseluruhan program ditulis dalam bahasa C. Stuktur data matriks terdapat pada file matrix.h dan matrix.c, sementara struktur data list berkait terdapat pada file wordlist.h dan wordlist.c. Matriks pada program digunakan untuk merepresentasikan *puzzle* yang ada, dan list berkait digunakan untuk menyimpan daftar kata-kata yang harus dicari pada permainan.

Pada matriks, elemen yang disimpan merupakan tipe data bentukan yang adalah gabungan dari karakter dan integer sebagai kode dari warna karakter. Definisi serta fungsi struktur data matriks yang tertera pada file matrix.h adalah seperti sebagai berikut.

```c
#ifndef MATRIX_H
#define MATRIX_H

#include <stdio.h>

typedef struct eltype {
    char letter;
    int color_code;
} eltype;

/* COLOR CODE
0 - default color
1 - red
```

```
2 - green
3 - yellow
4 - blue
5 - purple
6 - cyan */

typedef struct matrix {
    eltype buffer[50][50];
    int nRow;
    int nCol;
} matrix;

#define elmt(M,i,j) (M).buffer[i][j].letter
#define color(M,i,j) (M).buffer[i][j].color_code
#define row(M) (M).nRow
#define col(M) (M).nCol

void copy_matrix (matrix m1, matrix *m2);
void print_matrix (matrix m);

#endif
```

Realisasi dari fungsi dan metode struktur data matriks ditulis dalam file matrix.c.

```
#include "matrix.h"

void copy_matrix (matrix m1, matrix *m2) {
    int i, j;
    for (i = 0; i < 50; i++) {
        for (j = 0; j < 50; j++) {
            elmt(*m2,i,j) = elmt(m1,i,j);
            color(*m2,i,j) = color(m1,i,j);
        }
    }
    row(*m2) = row(m1);
    col(*m2) = col(m1);
}

void print_matrix (matrix m) {
    int i;
    int j;
    for (i = 0; i < row(m); i++) {
        for (j = 0; j < col(m); j++) {
            if (color(m,i,j) == 1) {
                printf("\033[0;31m");
            }
            else if (color(m,i,j) == 2) {
                printf("\033[0;32m");
```

```
            }
            else if (color(m,i,j) == 3) {
                printf("\033[0;33m");
            }
            else if (color(m,i,j) == 4) {
                printf("\033[0;34m");
            }
            else if (color(m,i,j) == 5) {
                printf("\033[0;35m");
            }
            else if (color(m,i,j) == 6) {
                printf("\033[0;36m");
            }
            printf("%c ", elmt(m,i,j));
            printf("\033[0m");
        }
        printf("\n");
    }
}
```

Sementara itu, pada struktur *list* berkait setiap elemennya menyimpan informasi berupa string, bilangan bulat yang menyatakan panjang string, serta alamat dari simpul berikutnya. Untuk menghasilkan pencarian yang sistematis, daftar kata diperlakukan seperti sebuah antrian. Definisi, selektor, serta fungsi dari struktur ini terdapat pada file wordlist.h.

```
#ifndef WORDLIST_H
#define WORDLIST_H

#include <stdio.h>
#include <stdlib.h>
#include "boolean.h"

typedef struct tnode* Address;
typedef struct tnode {
    char word[20];
    int length;
    Address next;
} Node;
typedef Address wordList;
#define word(L) (L)->word
#define next(L) (L)->next
#define first(L) (L)
#define length(L) (L)->length

Address newNode (char w[20]);
void enqueue (wordList *wl, char w[20]);
void dequeue (wordList *wl, wordList *wOut);
void print_wordList (wordList w);
```

```
#endif
```

Realisasi dari struktur data *list* tercantum pada file wordlist.c.

```
#include "wordlist.h"

Address newNode (char w[20]) {
    int n;
    Address p = (Address) malloc(sizeof(Node));
    if (p != NULL) {
        n = 0;
        while (n < 20 && (int) w[n] >= 65 && (int) w[n] <= 90) {
            word(p)[n] = w[n];
            n++;
        }
        length(p) = n;
        next(p) = NULL;
    }
    return p;
}

void enqueue (wordList *wl, char w[20]) {
    Address pNew, pList;
    pNew = newNode(w);
    if (pNew != NULL) {
        if (first(*wl) == NULL) {
            first(*wl) = pNew;
        }
        else {
            pList = first(*wl);
            while (next(pList) != NULL) {
                pList = next(pList);
            }
            next(pList) = pNew;
        }
    }
}

void dequeue (wordList *wl, wordList *wOut) {
    Address p;
    p = first(*wl);
    if (next(p) == NULL) {
        first(*wl) = NULL;
    }
    else {
        first(*wl) = next(p);
    }
```

```
    enqueue(wOut, word(p));
    free(p);
}

void print_wordList (wordList w) {
    Address p = first(w);
    int i;
    while (p != NULL) {
        for (i = 0; i < length(p); i++) {
            printf("%c", word(p)[i]);
        }
        printf("\n");
        p = next(p);
    }
}
```

Program utama sendiri dapat dibagi menjadi tiga bagian utama, yaitu pembacaan dan pengolahan file, pencarian kata, dan pengolahan serta penampilan matriks jawaban. Pembacaan file dilakukan oleh satu prosedur yaitu read_file. Pencarian kata dengan memanfaatkan rekursi dilakukan oleh fungsi check_N, check_S, check_E, check_W, check_NE, check_SE, check_SW, dan check_NW. Pengolahan matriks jawaban terdapat pada prosedur ans_matrix. Untuk menghitung waktu pencarian kata, digunakan prosedur clock_gettime yang berada pada file header sys/time.h.

```
#include <stdio.h>
#include <sys/time.h>
#include <stdlib.h>
#include <string.h>
#include "boolean.h"
#include "matrix.h"
#include "wordlist.h"

void read_file (char *file_name, matrix *m, wordList *wl) {
    // membaca file
    char dir[] = "./test/";
    char ch;
    boolean read_grid;
    int i, j, k;
    char wrd[20];

    strncat(dir, file_name, 20);
    FILE *fp = fopen(dir, "r");

    if (fp == NULL) {
        printf("Fail to read the file!\n");
        row(*m) = 0;
        col(*m) = 0;
    }
```

```
    else {
        ch = fgetc(fp);
        read_grid = true;

        i = 0;
        j = 0;

        for (k = 0; k < 20; k++) {
            wrd[k] = '\0';
        }
        k = 0;

        while (ch != EOF) {

            if (read_grid) {
                if (j == 0 && ch == '\n') {
                    read_grid = false; // found a blank line
                }
                else if (ch == '\n') {
                    // go to the next row
                    i++;
                    row(*m) = i;
                    j = 0;
                }
                else {
                    if ((int) ch != 32) {
                        elmt(*m,i,j) = ch;
                        color(*m,i,j) = 0;
                        j++;
                        col(*m) = j;
                    }
                }
            }

            else {
                if (ch == '\n') {
                    enqueue(wl, wrd);
                    for (k = 0; k < 20; k++) {
                        wrd[k] = '\0';
                    }
                    k = 0;
                }
                else {
                    wrd[k] = ch;
                    k++;
                }
            }
            ch = fgetc(fp);
```

```
        }
        enqueue(wl, wrd);
        fclose(fp);
    }
}

boolean check_N (wordList w, int k, matrix m, int i, int j) {
    // memeriksa ke arah utara
    if (k == length(w)) {
        return true;
    }
    else if (i == -1 || word(w)[k] != elmt(m,i,j)) {
        return false;
    }
    else {
        return (check_N(w, k+1, m, i-1, j));
    }
}

boolean check_S (wordList w, int k, matrix m, int i, int j) {
    // memeriksa ke arah selatan
    if (k == length(w)) {
        return true;
    }
    else if (i == row(m) || word(w)[k] != elmt(m,i,j)) {
        return false;
    }
    else {
        return (check_S(w, k+1, m, i+1, j));
    }
}

boolean check_W (wordList w, int k, matrix m, int i, int j) {
    // memeriksa ke arah barat
    if (k == length(w)) {
        return true;
    }
    else if (j == -1 || word(w)[k] != elmt(m,i,j)) {
        return false;
    }
    else {
        return (check_W(w, k+1, m, i, j-1));
    }
}

boolean check_E (wordList w, int k, matrix m, int i, int j) {
    // memeriksa ke arah timur
    if (k == length(w)) {
```

```
        return true;
    }
    else if (j == col(m) || word(w)[k] != elmt(m,i,j)) {
        return false;
    }
    else {
        return (check_E(w, k+1, m, i, j+1));
    }
}

boolean check_NE (wordList w, int k, matrix m, int i, int j) {
    // memeriksa ke arah timur laut
    if (k == length(w)) {
        return true;
    }
    else if (i == -1 ||j == col(m) || word(w)[k] != elmt(m,i,j)) {
        return false;
    }
    else {
        return (check_NE(w, k+1, m, i-1, j+1));
    }
}

boolean check_SE (wordList w, int k, matrix m, int i, int j) {
    // memeriksa ke arah tenggara
    if (k == length(w)) {
        return true;
    }
    else if (i == row(m) || j == col(m) || word(w)[k] != elmt(m,i,j)) {
        return false;
    }
    else {
        return (check_SE(w, k+1, m, i+1, j+1));
    }
}

boolean check_SW (wordList w, int k, matrix m, int i, int j) {
    // memeriksa ke arah barat daya
    if (k == length(w)) {
        return true;
    }
    else if (i == row(m) || j == -1 || word(w)[k] != elmt(m,i,j)) {
        return false;
    }
    else {
        return (check_SW(w, k+1, m, i+1, j-1));
    }
}
```

```
boolean check_NW (wordList w, int k, matrix m, int i, int j) {
    // memeriksa ke arah barat laut
    if (k == length(w)) {
        return true;
    }
    else if (i == -1 || j == -1 || word(w)[k] != elmt(m,i,j)) {
        return false;
    }
    else {
        return (check_NW(w, k+1, m, i-1, j-1));
    }
}

void ans_matrix (wordList w, int i, int j, matrix *m_ans, char dir) {
    // mengupdate matriks yang berisi kumpulan jawaban dengan memberi warna
pada kata yang telah ditemukan
    int x, y, l;
    int rand_color;

    rand_color = (rand() % 6) + 1;

    x = i;
    y = j;
    l = 0;

    if (dir == 'N') { //north
        while (l < length(w)) {
            if (color(*m_ans, x, y) == rand_color) {
                rand_color = (rand() % 6) + 1;
                l = 0;
                x = i;
                y = j;
            }
            else {
                color(*m_ans, x, y) = rand_color;
                l++;
                x--;
            }
        }
    }
    else if (dir == 'S') { //south
        while (l < length(w)) {
            if (color(*m_ans, x, y) == rand_color) {
                rand_color = (rand() % 6) + 1;
                l = 0;
                x = i;
                y = j;
```

```
            }
            else {
                color(*m_ans, x, y) = rand_color;
                l++;
                x++;
            }
        }
    }
    else if (dir == 'E') { //east
        while (l < length(w)) {
            if (color(*m_ans, x, y) == rand_color) {
                rand_color = (rand() % 6) + 1;
                l = 0;
                x = i;
                y = j;
            }
            else {
                color(*m_ans, x, y) = rand_color;
                l++;
                y++;
            }
        }
    }
    else if (dir == 'W') { //west
        while (l < length(w)) {
            if (color(*m_ans, x, y) == rand_color) {
                rand_color = (rand() % 6) + 1;
                l = 0;
                x = i;
                y = j;
            }
            else {
                color(*m_ans, x, y) = rand_color;
                l++;
                y--;
            }
        }
    }
    else if (dir == 'U') { //northeast
        while (l < length(w)) {
            if (color(*m_ans, x, y) == rand_color) {
                rand_color = (rand() % 6) + 1;
                l = 0;
                x = i;
                y = j;
            }
            else {
                color(*m_ans, x, y) = rand_color;
```

```
                l++;
                x--;
                y++;
            }
        }
    }
    else if (dir == 'M') { //southeast
        while (l < length(w)) {
            if (color(*m_ans, x, y) == rand_color) {
                rand_color = (rand() % 6) + 1;
                l = 0;
                x = i;
                y = j;
            }
            else {
                color(*m_ans, x, y) = rand_color;
                l++;
                x++;
                y++;
            }
        }
    }
    else if (dir == 'B') { //southwest
        while (l < length(w)) {
            if (color(*m_ans, x, y) == rand_color) {
                rand_color = (rand() % 6) + 1;
                l = 0;
                x = i;
                y = j;
            }
            else {
                color(*m_ans, x, y) = rand_color;
                l++;
                x++;
                y--;
            }
        }
    }
    else if (dir == 'T') { //northwest
        while (l < length(w)) {
            if (color(*m_ans, x, y) == rand_color) {
                rand_color = (rand() % 6) + 1;
                l = 0;
                x = i;
                y = j;
            }
            else {
                color(*m_ans, x, y) = rand_color;
```

```
                l++;
                x--;
                y--;
            }
        }
    }
}

int main () {
    matrix m, m_ans;
    wordList l, l_notfound, l_found;
    char filename[20];
    int i, j, k, g, o;
    boolean found;
    struct timespec begin, end;
    float searchtime;

    // membaca file
    printf("Enter file name : ");
    scanf("%s", &filename);
    read_file(filename, &m, &l);

    if (row(m) != 0 || col(m) != 0) {


        copy_matrix(m, &m_ans); // membuat matriks jawaban

        printf("matrix size is : %dx%d\n\n", row(m), col(m));

        while (l != NULL) {
            // begin the brute force for each word in the list
            found = false;
            i = 0;
            j = 0;

            // begin time count
            clock_gettime(CLOCK_MONOTONIC, &begin);

            while (!found && i < row(m)) {
                if (j == col(m)) {
                    // ke baris berikutnya
                    i++;
                    j = 0;
                }
                else if (check_N(first(l), 0, m, i, j)) {
                    clock_gettime(CLOCK_MONOTONIC, &end); // stop time count
                    ans_matrix(first(l), i, j, &m_ans, 'N');
                    found = true;
                }
```

```
                else if (check_S(first(l), 0, m, i, j)) {
                    clock_gettime(CLOCK_MONOTONIC, &end); // stop time count
                    ans_matrix(first(l), i, j, &m_ans, 'S');
                    found = true;
                }
                else if (check_E(first(l), 0, m, i, j)) {
                    clock_gettime(CLOCK_MONOTONIC, &end); // stop time count
                    ans_matrix(first(l), i, j, &m_ans, 'E');
                    found = true;
                }
                else if (check_W(first(l), 0, m, i, j)) {
                    clock_gettime(CLOCK_MONOTONIC, &end); // stop time count
                    ans_matrix(first(l), i, j, &m_ans, 'W');
                    found = true;
                }
                else if (check_NE(first(l), 0, m, i, j)) {
                    clock_gettime(CLOCK_MONOTONIC, &end); // stop time count
                    ans_matrix(first(l), i, j, &m_ans, 'U');
                    found = true;
                }
                else if (check_SE(first(l), 0, m, i, j)) {
                    clock_gettime(CLOCK_MONOTONIC, &end); // stop time count
                    ans_matrix(first(l), i, j, &m_ans, 'M');
                    found = true;
                }
                else if (check_SW(first(l), 0, m, i, j)) {
                    clock_gettime(CLOCK_MONOTONIC, &end); // stop time count
                    ans_matrix(first(l), i, j, &m_ans, 'B');
                    found = true;
                }
                else if (check_NW(first(l), 0, m, i, j)) {
                    clock_gettime(CLOCK_MONOTONIC, &end); // stop time count
                    ans_matrix(first(l), i, j, &m_ans, 'T');
                    found = true;
                }
                else {
                    // ke kolom berikutnya
                    j++;
                }
            }
            if (found) {
                searchtime = (end.tv_nsec - begin.tv_nsec) / 1000000000.0 +
(end.tv_sec  - begin.tv_sec);

                printf("word ");
                for (g = 0; g < length(first(l)); g++) {
                    printf("%c", word(first(l))[g]);
                }
```

```
                printf(" found in %f seconds\n", searchtime);
                dequeue(&l, &l_found);
            }
            else {
                dequeue(&l, &l_notfound);
            }
        }
        printf("\n");
        print_matrix(m_ans);
        printf("\n");
        if (l_notfound != NULL) {
            printf("These are the words which cannot be found :\n");
            print_wordList(l_notfound);
        }
        printf("\n");
        printf("Fin\n");
    }
    return 0;
}
```

III. HASIL PERCOBAAN DAN EVALUASI

Terdapat sembilan *puzzle* dengan tiga ukuran berbeda yang digunakan untuk melakukan percobaan pada program. Ukuran *puzzle* dibagi menjadi tiga jenis, yaitu *small*, *medium*, dan *large*. Hasil percobaan terdapat pada Tabel 3.1. Pada beberapa percobaan, tangkapan layar dari keluaran program dipotong karena ukuran matriks yang cukup besar dan jumlah kata yang cukup banyak.

Tabel 3.1. Hasil Percobaan pada Program

Nama file	Isi file	Hasil pencarian oleh program
small1.txt	A K P H O E N I X A M D I R L R R L A Q V Q H S I N R O C U T U G D D A N C S N Q Y R M C A Y S N N I H G W A W O E S O M B P U A V A A V T G E N N C U V M A L A Y Z D W K S I A H S P I W L T O W Y A S N P R I M J C N O O A T B J R R M A N I A W X H R P N N O W X A M O G I A V O J S J H V X D C R G U A N O C N A G A E L E E S A P M X S N O E G S V E V L P V J L W I Y L Y S Y O A R G G A V M O C B R U Y B D A C P E B F D J P U S T B U C K E Y E L ANCONA CORNISH MARANS ASEEL HOLLAND PHOENIX BUCKEYE JAVA SULTAN CAMPINE LEGHORN SUMATRA COCHIN MALAY SUSSEX	Enter file name : small1.txt matrix size is : 15x15 word ANCONA found in 0.001047 seconds word CORNISH found in 0.000154 seconds word MARANS found in 0.000632 seconds word ASEEL found in 0.000812 seconds word HOLLAND found in 0.000549 seconds word PHOENIX found in 0.000017 seconds word BUCKEYE found in 0.000947 seconds word JAVA found in 0.000624 seconds word SULTAN found in 0.000220 seconds word CAMPINE found in 0.000676 seconds word LEGHORN found in 0.000745 seconds word SUMATRA found in 0.000968 seconds word COCHIN found in 0.000261 seconds word MALAY found in 0.000363 seconds word SUSSEX found in 0.000740 seconds A K P H O E N I X A M D I R L R R L A Q V Q H S I N R O C U T U G D D A N C S N Q Y R M C A Y S N N I H G W A W O E S O M B P U A V A A V T G E N N C U V M A L A Y Z D W K S I A H S P I W L T O W Y A S N P R I M J C N O O A T B J R R M A N I A W X H R P N N O W X A M O G I A V O J S J H V X D C R G U A N O C N A G A E L E E S A P M X S N O E G S V E V L P V J L W I Y L Y S Y O A R G G A V M O C B R U Y B D A C P E B F D J P U S T B U C K E Y E L

small2.txt

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

BLIND
HOUND
SLITEYE
BLUE
LEMON
TIGER
BULL
MILK
WEASEL
CAT
PIGEYE
WHITE
FOSSIL
SILKY
ZEBRA
```

```
Enter file name : small2.txt
matrix size is : 15x15

word BLIND found in 0.000450 seconds
word HOUND found in 0.001343 seconds
word SLITEYE found in 0.000768 seconds
word BLUE found in 0.001088 seconds
word LEMON found in 0.002166 seconds
word TIGER found in 0.000180 seconds
word BULL found in 0.000933 seconds
word MILK found in 0.000326 seconds
word WEASEL found in 0.000360 seconds
word CAT found in 0.000087 seconds
word PIGEYE found in 0.000884 seconds
word WHITE found in 0.001029 seconds
word FOSSIL found in 0.000209 seconds
word SILKY found in 0.000027 seconds
word ZEBRA found in 0.000696 seconds

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

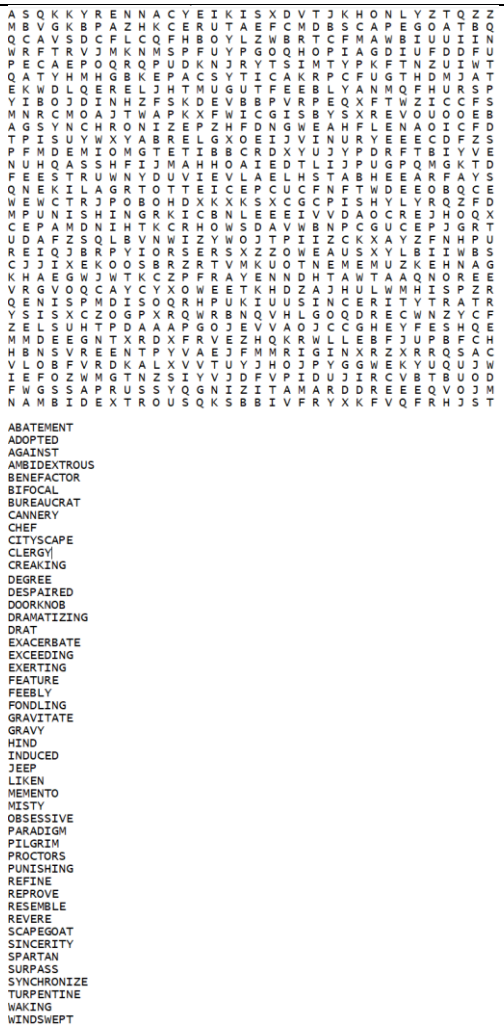
small3.txt	W H U T M U R K V F U Z Z N H A E N Y J G O E N L B P O N Z U Y R V F O W K T W M P G R M T Y S C H I F N B S I D B E L B I E R S C E F K L U J M D T A I A X N R R L C Q H L B A D R E H C X Z A Z G U B L C E Q B G K U B N K E M N N H S R E E F W V G K A K F H A R B H B L S F A C G A R D A E D O T S L M T U W W Y H O V S O L Z X T H B U C U W A E J P E R H U B B D O D R E L A S A H P T J R P B R T S A Z E B A B K A W E N V S J O N E E B O L R A E BAJORAN BARBELL CLIPON CLUSTER DANGLE EARCUFF EARHOOK EARSCREW EARLOBE HOOPS HUGGY JHUMKA LEVERS STUDS THREADER	Enter file name : small3.txt matrix size is : 15x15 word BAJORAN found in 0.001798 seconds word BARBELL found in 0.001411 seconds word CLIPON found in 0.000694 seconds word CLUSTER found in 0.000622 seconds word DANGLE found in 0.001079 seconds word EARCUFF found in 0.001213 seconds word EARHOOK found in 0.000536 seconds word EARSCREW found in 0.000697 seconds word EARLOBE found in 0.002004 seconds word HOOPS found in 0.000829 seconds word HUGGY found in 0.000334 seconds word JHUMKA found in 0.000432 seconds word LEVERS found in 0.001838 seconds word STUDS found in 0.000821 seconds word THREADER found in 0.001627 seconds W H U T M U R K V F U Z Z N H A E N Y J G O E N L B P O N Z U Y R V F O W K T W M P G R M T Y S C H I F N B S I D B E L B I E R S C E F K L U J M D T A I A X N R R L C Q H L B A D R E H C X Z A Z G U B L C E Q B G K U B N K E M N N H S R E E F W V G K A K F H A R B H B L S F A C G A R D A E D O T S L M T U W W Y H O V S O L Z X T H B U C U W A E J P E R H U B B D O D R E L A S A H P T J R P B R T S A Z E B A B K A W E N V S J O N E E B O L R A E

medium1.txt	E X L L S T F X S H S P P J Q F I Q R H J W W B R B Z S U S U O T Z Z T J M P R W X N E G S K P S R W D N G O M S B G B T E R C E J P A C M S A N X S X B O E C I T S L O S B M W B L U E C L I P S E P G T T E N A L P U B E X E A D G V C O E L W T O E R E R A Y L W Y V W H X B F S T C B E F N J O G X H P A W B W K B A Y A A S L B R A L B H R W S O P K B R J G P N Y N I I P G I S X B X Z G R Q V X X X I L E D G U P F O O M L O Z P N K U Q R A E Y T H G I L T H H B U N T B M I G R K Z Y J E R E D I R I E V N I L K V E X N T G R R O A X M K J G C Y A U R E D F R A B M X E N H E Q U D B G P R Q L W R T H U W E Z W S Z Z R B W N B D N E R V L I B O Y X Q E Q N L C M P F X D X E I Y E I L Q Y Y H V N M E S Z J O S V K U L B N L P N N Q D O K I T D X F L S Z G Y W E U B A K M O G P R C T E Z Y U F H U O B M M R T Q W Q A D K L Z H M H P I P H A S E S R P I Z F D V T W Z T A C ECLIPSE LUNAR SOLSTICE ECLIPTIC METEOR SUNSPOT EQUINOX NEBULA WANING GALAXY PHASE WAXING LIGHTYEAR PLANET ZENITH	word ECLIPSE found in 0.000791 seconds word LUNAR found in 0.001363 seconds word SOLSTICE found in 0.001612 seconds word ECLIPTIC found in 0.000827 seconds word METEOR found in 0.000381 seconds word SUNSPOT found in 0.000090 seconds word EQUINOX found in 0.001514 seconds word NEBULA found in 0.000322 seconds word WANING found in 0.001828 seconds word GALAXY found in 0.001864 seconds word PHASE found in 0.002522 seconds word WAXING found in 0.001314 seconds word LIGHTYEAR found in 0.001566 seconds word PLANET found in 0.000414 seconds word ZENITH found in 0.002015 seconds E X L L S T F X S H S P P J Q F I Q R H J W W B R B Z S U S U O T Z Z T J M P R W X N E G S K P S R W D N G O M S B G B T E R C E J P A C M S A N X S X B O E C I T S L O S B M W B L U E C L I P S E P G T T E N A L P U B E X E A D G V C O E L W T O E R E R A Y L W Y V W H X B F S T C B E F N J O G X H P A W B W K B A Y A A S L B R A L B H R W S O P K B R J G P N Y N I I P G I S X B X Z G R Q V X X X I L E D G U P F O O M L O Z P N K U Q R A E Y T H G I L T H H B U N T B M I G R K Z Y J E R E D I R I E V N I L K V E X N T G R R O A X M K J G C Y A U R E D F R A B M X E N H E Q U D B G P R Q L W R T H U W E Z W S Z Z R B W N B D N E R V L I B O Y X Q E Q N L C M P F X D X E I Y E I L Q Y Y H V N M E S Z J O S V K U L B N L P N N Q D O K I T D X F L S Z G Y W E U B A K M O G P R C T E Z Y U F H U O B M M R T Q W Q A D K L Z H M H P I P H A S E S R P I Z F D V T W Z T A C

medium2.txt	D P V X N P W O V S S E G R A S J J H F Y P Q K J N F J V Y I N G A V G D S M B V C S W P I G L H G B T R E I K B Y U B B Y B T F U Z Z H H Y D T Z O X S U R O L H F N I P I V Y O U S J O B Y X I B J N R H B T C L A I U W O U C U N U E S H Q I X D E E K Z C R J X Y W O N S S T E H J T I M X U Y J A G B A L X M F D O I H A C U B T F O Z G K D S O P U A U E F H T O C L I W A Q F O E Q M E I A L V T Y W M Z L G O B V M D O J R S E I L N U H U T H E Y J N O C I S S U P A G Z L E E R W C P A Z Y W R S C L A C N L D S O V D S G W E P G M O U K A L W B V L U C N M X X E Q S H M N R E J X S Q E J G J K G H P I R S T G O U B J Q S E S O L K O J G Y K A A N X J F M L M S L W O F P W E E R A B A D M X E J X K K C A L B X R Z L Z A U F M F L Q P E J E N I A R G T R O H S I O G Z Z B I F F G V V A L E N C I A Y P N D J J P F X W O K N Z F H Z U O Q M B A R L E Y Q N W U F K F H M BASMATI BLACK BROWN GLUTINOUS JAPANESE JASMINE LONGGRAIN PARBOILED ROSEMATTA SHORTGRAIN STICKY SUSHI VALENCIA WHITE WILD	word BASMATI found in 0.001588 seconds word BLACK found in 0.002038 seconds word BROWN found in 0.003170 seconds word GLUTINOUS found in 0.001032 seconds word JAPANESE found in 0.000645 seconds word JASMINE found in 0.000943 seconds word LONGGRAIN found in 0.000969 seconds word PARBOILED found in 0.000584 seconds word ROSEMATTA found in 0.002720 seconds word SHORTGRAIN found in 0.002257 seconds word STICKY found in 0.000471 seconds word SUSHI found in 0.000991 seconds word VALENCIA found in 0.003171 seconds word WHITE found in 0.000922 seconds word WILD found in 0.002683 seconds D P V X N P W O V S S E G R A S J J H F Y P Q K J N F J V Y I N G A V G D S M B V C S W P I G L H G B T R E I K B Y U B B Y B T F U Z Z H H Y D T Z O X S U R O L H F N I P I V Y O U S J O B Y X I B J N R H B T C L A I U W O U C U N U E S H Q I X D E E K Z C R J X Y W O N S S T E H J T I M X U Y J A G B A L X M F D O I H A C U B T F O Z G K D S O P U A U E F H T O C L I W A Q F O E Q M E I A L V T Y W M Z L G O B V M D O J R S E I L N U H U T H E Y J N O C I S S U P A G Z L E E R W C P A Z Y W R S C L A C N L D S O V D S G W E P G M O U K A L W B V L U C N M X X E Q S H M N R E J X S Q E J G J K G H P I R S T G O U B J Q S E S O L K O J G Y K A A N X J F M L M S L W O F P W E E R A B A D M X E J X K K C A L B X R Z L Z A U F M F L Q P E J E N I A R G T R O H S I O G Z Z B I F F G V V A L E N C I A Y P N D J J P F X W O K N Z F H Z U O Q M B A R L E Y Q N W U F K F H M

medium3.txt	K L R Z I Q N H E G W O W N P M D N E Q N D G N I H T E M O S A H D S L S P I R Z N H L D H V H A A P H I T S A W M O W T A O X A T Q O X F L E T R O C L D K U B S Z E H W X O Z K A D T T D O I I O A F B V T H E M A R K N H N Y A N T P N D I M M B T A V Q K E X T U A E I N P S E R Q E T M N E Z V P I I L R G E E O L I X G C S U I I A Y N W G O A R E X D C G W M E C N C K O C Y C E H C I L C A N A D X V T G O E T P Z I B R R Z W F M M S B Q N Z E U S H S E G K F Y O J N G O P O U Y W P R E R T B H U P V P O P O G Z D J D R T N E X E T B T Z A W E B T J E Y N B U E E Z B J P P U D K U A G N O I T C A N I J M U M F N A V O D K L R F L W A M C J Z E P E I B L I M Q L X F A Y O U X W A L T U O Q I B J E A B K P A R T S N V V W C O R F L U L W N I B C Y L I N D E R P A P Y O U L W K S Q D L U I G N L B T O Q G D Q R C K T Y A Z C P P I P U M O B L F A F C K X O W V E I A Q COMMA CYLINDER INACTION LINK OUT PARTS PERSONS SOMETHING THECUT THEMARK THEPOINT TOOTH TREASURE WORD YOU	word COMMA found in 0.000522 seconds word CYLINDER found in 0.003140 seconds word INACTION found in 0.001904 seconds word LINK found in 0.001016 seconds word OUT found in 0.001839 seconds word PARTS found in 0.002160 seconds word PERSONS found in 0.001650 seconds word SOMETHING found in 0.000211 seconds word THECUT found in 0.001013 seconds word THEMARK found in 0.000571 seconds word THEPOINT found in 0.001698 seconds word TOOTH found in 0.000600 seconds word TREASURE found in 0.000865 seconds word WORD found in 0.000429 seconds word YOU found in 0.001209 seconds K L R Z I Q N H E G W O W N P M D N E Q N D G N I H T E M O S A H D S L S P I R Z N H L D H V H A A P H I T S A W M O W T A O X A T Q O X F L E T R O C L D K U B S Z E H W X O Z K A D T T D O I I O A F B V T H E M A R K N H N Y A N T P N D I M M B T A V Q K E X T U A E I N P S E R Q E T M N E Z V P I I L R G E E O L I X G C S U I I A Y N W G O A R E X D C G W M E C N C K O C Y C E H C I L C A N A D X V T G O E T P Z I B R R Z W F M M S B Q N Z E U S H S E G K F Y O J N G O P O U Y W P R E R T B H U P V P O P O G Z D J D R T N E X E T B T Z A W E B T J E Y N B U E E Z B J P P U D K U A G N O I T C A N I J M U M F N A V O D K L R F L W A M C J Z E P E I B L I M Q L X F A Y O U X W A L T U O Q I B J E A B K P A R T S N V V W C O R F L U L W N I B C Y L I N D E R P A P Y O U L W K S Q D L U I G N L B T O Q G D Q R C K T Y A Z C P P I P U M O B L F A F C K X O W V E I A Q

large1.txt



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

ABATEMENT
ADOPTED
AGAINST
AMBIDEXTROUS
BENEFACTOR
BIFOCAL
BUREAUCRAT
CANNERY
CHEF
CITYSCAPE
CLERGY
CREAKING
DEGREE
DESPAIRED
DOORKNOB
DRAMATIZING
DRAT
EXACERBATE
EXCEEDING
EXERTING
FEATURE
FEEBLY
FONDLING
GRAVITATE
GRAVY
HIND
INDUCED
JEEP
LIKEN
MEMENTO
MISTY
OBSESSIVE
PARADIGM
PILGRIM
PROCTORS
PUNISHING
REFINE
REPROVE
RESEMBLE
REVERE
SCAPEGOAT
SINCERITY
SPARTAN
SURPASS
SYNCHRONIZE
TURPENTINE
WAKING
WINDSWEPT
```

```
Enter file name : large1.txt
matrix size is : 32x34

word ABATEMENT found in 0.000261 seconds
word ADOPTED found in 0.005441 seconds
word AGAINST found in 0.004229 seconds
word AMBIDEXTROUS found in 0.004281 seconds
word BENEFACTOR found in 0.001698 seconds
word BIFOCAL found in 0.001035 seconds
word BUREAUCRAT found in 0.002843 seconds
word CANNERY found in 0.000098 seconds
word CHEF found in 0.005719 seconds
word CITYSCAPE found in 0.002750 seconds
word CLERGY found in 0.004182 seconds
word CREAKING found in 0.000447 seconds
word DEGREE found in 0.002145 seconds
word DESPAIRED found in 0.003223 seconds
word DOORKNOB found in 0.001664 seconds
word DRAMATIZING found in 0.005412 seconds
word DRAT found in 0.000498 seconds
word EXACERBATE found in 0.004104 seconds
word EXCEEDING found in 0.003313 seconds
word EXERTING found in 0.001019 seconds
word FEATURE found in 0.000292 seconds
word FEEBLY found in 0.001658 seconds
word FONDLING found in 0.006236 seconds
word GRAVITATE found in 0.004578 seconds
word GRAVY found in 0.005061 seconds
word HIND found in 0.002322 seconds
word INDUCED found in 0.000667 seconds
word JEEP found in 0.001607 seconds
word LIKEN found in 0.002531 seconds
word MEMENTO found in 0.003381 seconds
word MISTY found in 0.000725 seconds
word OBSESSIVE found in 0.007344 seconds
word PARADIGM found in 0.003379 seconds
word PILGRIM found in 0.005137 seconds
word PROCTORS found in 0.005072 seconds
word PUNISHING found in 0.003271 seconds
word REFINE found in 0.002709 seconds
word REPROVE found in 0.002960 seconds
word RESEMBLE found in 0.004666 seconds
word REVERE found in 0.005850 seconds
word SCAPEGOAT found in 0.000232 seconds
word SINCERITY found in 0.003772 seconds
word SPARTAN found in 0.003610 seconds
word SURPASS found in 0.005338 seconds
word SYNCHRONIZE found in 0.001470 seconds
word TURPENTINE found in 0.005207 seconds
word WAKING found in 0.003511 seconds
word WINDSWEPT found in 0.003971 seconds

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

large2.txt

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

ACCEPTABLE
DECEIVER
LAVENDER
SLEEP
ARCHERY
DISGRACEFUL
LEVELED
SOAP
ASSOCIATED
DIVIDING
MESSIAH
SPATIALLY
ATTACHMENT
EARTHQUAKE
NEWSBOY
STEPSISTER
AUDACIOUS
EYEFUL
ORPHAN
STIRRING
BANGKOK
FIESTA
PILOTS
STORIES
BOORISHNESS
FIREWOOD
PORES
SUBJECTIVE
BRITTLE
GABLE
PREVALENCE
SURPRISING
CAUTIOUS
HINDRANCE
PUBLICIST
TONER
CELERY
IMAGO
RESTRICTIVE
UNLEARN
CONSORTED
INCEPTION
RETIRING
VOLE
CREATURE
KEYNOTE
SEIZURE
WRONGDOING
```

```
Enter file name : large2.txt
matrix size is : 32x34

word ACCEPTABLE found in 0.004628 seconds
word DECEIVER found in 0.000346 seconds
word LAVENDER found in 0.003648 seconds
word SLEEP found in 0.002566 seconds
word ARCHERY found in 0.004447 seconds
word DISGRACEFUL found in 0.003024 seconds
word LEVELED found in 0.000099 seconds
word SOAP found in 0.002088 seconds
word ASSOCIATED found in 0.002191 seconds
word DIVIDING found in 0.001333 seconds
word MESSIAH found in 0.001984 seconds
word SPATIALLY found in 0.005887 seconds
word ATTACHMENT found in 0.006766 seconds
word EARTHQUAKE found in 0.004645 seconds
word NEWSBOY found in 0.005449 seconds
word STEPSISTER found in 0.005647 seconds
word AUDACIOUS found in 0.002981 seconds
word EYEFUL found in 0.000252 seconds
word ORPHAN found in 0.004406 seconds
word STIRRING found in 0.002118 seconds
word BANGKOK found in 0.004509 seconds
word FIESTA found in 0.002392 seconds
word PILOTS found in 0.002255 seconds
word STORIES found in 0.000480 seconds
word BOORISHNESS found in 0.004011 seconds
word FIREWOOD found in 0.003437 seconds
word PORES found in 0.002838 seconds
word SUBJECTIVE found in 0.000171 seconds
word BRITTLE found in 0.003718 seconds
word GABLE found in 0.005534 seconds
word PREVALENCE found in 0.002198 seconds
word SURPRISING found in 0.002553 seconds
word CAUTIOUS found in 0.002339 seconds
word HINDRANCE found in 0.004131 seconds
word PUBLICIST found in 0.000358 seconds
word TONER found in 0.000212 seconds
word CELERY found in 0.004006 seconds
word IMAGO found in 0.004149 seconds
word RESTRICTIVE found in 0.003019 seconds
word UNLEARN found in 0.005050 seconds
word CONSORTED found in 0.003236 seconds
word INCEPTION found in 0.001245 seconds
word RETIRING found in 0.003934 seconds
word VOLE found in 0.003174 seconds
word CREATURE found in 0.002457 seconds
word KEYNOTE found in 0.004604 seconds
word SEIZURE found in 0.005716 seconds
word WRONGDOING found in 0.000334 seconds

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

large3.txt

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

```
AMALGAMATE
ELOPED
ONSHORE
SESSION
AMMONIA
FAITHFULLY
PACER
SISTER
ARISE
FRANKFURTER
PERSIA
SLAVISHLY
ASSET
GEARED
PINT
SNOB
BLUSHING
GNAWED
POKER
SPOOKY
CONJUGATING
HEAVEN
POLITEST
SYNERGY
CONVICTED
HOOD
REFINEMENT
TANG
DELVE
KEYPAD
SADSACK
UNIFY
DENIAL
LEVITY
SARONG
UNIFYING
DEVOUT
MACHINATION
SATIRIZED
UTAH
DOSING
MODULATION
SCHEDULED
VIDEO
EARLOBE
MOTIVATION
SEQUEL
ZEALOT
```

```
Enter file name : large3.txt
matrix size is : 32x34

word AMALGAMATE found in 0.001731 seconds
word ELOPED found in 0.003021 seconds
word ONSHORE found in 0.004658 seconds
word SESSION found in 0.004424 seconds
word AMMONIA found in 0.000392 seconds
word FAITHFULLY found in 0.003312 seconds
word PACER found in 0.002022 seconds
word SISTER found in 0.003478 seconds
word ARISE found in 0.004702 seconds
word FRANKFURTER found in 0.002515 seconds
word PERSIA found in 0.006223 seconds
word SLAVISHLY found in 0.000657 seconds
word ASSET found in 0.001965 seconds
word GEARED found in 0.003986 seconds
word PINT found in 0.001186 seconds
word SNOB found in 0.003422 seconds
word BLUSHING found in 0.002969 seconds
word GNAWED found in 0.000816 seconds
word POKER found in 0.001320 seconds
word SPOOKY found in 0.003488 seconds
word CONJUGATING found in 0.000510 seconds
word HEAVEN found in 0.002014 seconds
word POLITEST found in 0.002111 seconds
word SYNERGY found in 0.000211 seconds
word CONVICTED found in 0.000952 seconds
word HOOD found in 0.000748 seconds
word REFINEMENT found in 0.003879 seconds
word TANG found in 0.004579 seconds
word DELVE found in 0.003243 seconds
word KEYPAD found in 0.003758 seconds
word SADSACK found in 0.001265 seconds
word UNIFY found in 0.000761 seconds
word DENIAL found in 0.005115 seconds
word LEVITY found in 0.003354 seconds
word SARONG found in 0.001540 seconds
word UNIFYING found in 0.004971 seconds
word DEVOUT found in 0.002479 seconds
word MACHINATION found in 0.003228 seconds
word SATIRIZED found in 0.002792 seconds
word UTAH found in 0.002210 seconds
word DOSING found in 0.001121 seconds
word MODULATION found in 0.002900 seconds
word SCHEDULED found in 0.003789 seconds
word VIDEO found in 0.002027 seconds
word EARLOBE found in 0.002908 seconds
word MOTIVATION found in 0.004258 seconds
word SEQUEL found in 0.002423 seconds
word ZEALOT found in 0.004353 seconds

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

Berdasarkan hasil yang didapat, evaluasi untuk program dapat dirangkum dalam Tabel 3.2.

Tabel 3.2. Evaluasi Program

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

IV. ALAMAT KODE PROGRAM

Keseluruhan program dapat diakses pada repository GitHub dengan alamat :

https://github.com/chryes220/Tucil1_Stima.git