

Laporan Tugas Kecil 1
IF2211 Strategi Algoritma Semester II tahun 2021/2022
Penyelesaian Word Search Puzzle dengan Algoritma Brute Force



Yoseph Alexamder Siregar

K3 – 13520141

BAB 1

Algoritma Brute Force

Algoritma brute force yang digunakan pada pengerjaan tugas kecil 1 ini didasarkan pada konsep. algoritma bruteforce pada string / pattern matching. Wordpuzzle yang akan diolah diletakkan pada sebuah matriks dan daftar-daftar kata(keyword) yang akan dicari diletakkan pada sebuah list. Dilakukan iterasi pada list untuk tiap keyword dan juga iterasi matriks puzzle, diinisialisasi counter bernilai 0 (terdapat 8 counter untuk 8 arah pengecekan) , dilakukan pengecekan pada tiap elemen puzzle (huruf, dengan pengaksesan indeks yang berbeda berdasarkan arah pengecekan sehingga terdapat 8 loop untuk pengecekan tiap arah) terhadap tiap huruf dari keyword (diakses dengan indeks ke-counter). Apabila terdapat kecocokan, counter akan bertambah dan dilakukan pengecekan pada elemen matriks puzzle berikutnya (berdasarkan arah pengecekan) dan huruf berikutnya pada keyword. Setelah iterasi matriks puzzle sudah selesai, counter dari tiap arah akan dilakukan pengecekan terhadap panjang dari keyword. Apabila terdapat counter yang nilainya sama, maka didapatkan arah untuk keyword itu (note : indeks i dan j akan tetap berada pada huruf pertama dari keyword, misal JUPITER maka indeks i dan j akan tetap menunjuk pada elemen yang bernilai J). Setelah ditemukan arah dari keyword tersebut dari huruf pertamanya, akan diinisialisasi matriks berukuran sama dengan matriks puzzle dengan isi symbol '=' yang lalu akan digantikan pada elemen dengan indeks i dan j dari huruf pertama keyword dari matriks puzzle dan dilakukan iterasi untuk tiap huruf berikutnya dengan panduan arah yang didapatkan (untuk tiap arah yang berbeda, pengaksesan elemen dari matriks puzzle akan berbeda). Setelah elemen pada matriks baru sudah digantikan sesuai dengan keywordnya maka matriks tersebut akan ditampilkan sebagai output dari 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	✓	

BAB 2

Source Code Program

Source Code dari Tugas Kecil 1 pada bahasa C

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "boolean.h"
#include "time.h"

typedef char ElType;

typedef struct {
    ElType contents[99][99];
    int neff;
} List;
#define NEFF(L) (L).neff
#define LIST(L, i) (L).contents[(i)]

typedef struct {
    ElType contents[99][99];
    int rowEff;
    int colEff;
} Matrix;
#define ROW(M) (M).rowEff
#define COL(M) (M).colEff
#define MTRX(M, i, j) (M).contents[(i)][(j)]

int main(){
    char filename[99];
    printf("Masukkan nama file yang akan diinput: ");
    gets(filename);

    FILE * fileinput;
    Matrix puzzle;
    Matrix result;
    int row = 0;
    int col = 0;
    int count;
    char cc;
    char temp;
    ROW(puzzle) = 0;
```

```

int main(){
    char filename[99];
    printf("Masukkan nama file yang akan diinput: ");
    gets(filename);

    FILE * fileinput;
    Matrix puzzle;
    Matrix result;
    int row = 0;
    int col = 0;
    int count;
    char cc;
    char temp;
    ROW(puzzle) = 0;
    COL(puzzle) = 0;
    char *str;
    boolean newLine = false;

    fileinput=fopen(filename,"r");

    cc = fgetc(fileinput);

    while(!newLine){
        if(cc != ' ' && cc != '\n'){
            MTRX(puzzle, row, col) = cc;
            col++;
        }
        else if(cc == '\n'){
            ROW(puzzle)++;
            row++;
            COL(puzzle) = col;
            col = 0;
        }
        temp = cc;
        cc = fgetc(fileinput);
        if(cc == '\n' && temp == '\n'){
            newLine = true;
        }
    }
}

```

```

    }
}

List words;
NEFF(words) = 0;

char key[100];
while (fgets(key, sizeof(key), fileinput)) {
    strcpy(LIST(words, NEFF(words)), key);
    NEFF(words)++;
}

for (int i = 0; i < NEFF(words)-1; i++) {
    LIST(words, i)[strlen(LIST(words, i))-1] = '\\0';
}

fclose(fileinput);
count = 0;

clock_t beginpuzzle = clock();

for (int n = 0; n < NEFF(words); n++){
    int wordsLen = strlen(LIST(words,n));
    for (int i = 0; i < ROW(puzzle); i++){
        for (int j = 0; j < COL(puzzle); j++){
            count++;
            int m = 0; int o = 0; int p = 0; int q = 0;
            int r = 0; int s = 0; int t = 0; int u = 0;
            while ((m < wordsLen) && (MTRX(puzzle, i, j+m) == LIST(words, n)[m])){
                m++;
                count++;
            }
            while ((o < wordsLen) && (MTRX(puzzle, i+o, j) == LIST(words, n)[o])){
                o++;
                count++;
            }
        }
    }
}

```

```

while ((p<wordsLen)&&(MTRX(puzzle,i,j-p)==LIST(words,n)[p])){
    p++;
    count++;
}
while ((q<wordsLen)&&(MTRX(puzzle,i-q,j)==LIST(words,n)[q])){
    q++;
    count++;
}
while ((r<wordsLen)&&(MTRX(puzzle,i+r,j+r)==LIST(words,n)[r])){
    r++;
    count++;
}
while ((s<wordsLen)&&(MTRX(puzzle,i+s,j-s)==LIST(words,n)[s])){
    s++;
    count++;
}
while ((t<wordsLen)&&(MTRX(puzzle,i-t,j-t)==LIST(words,n)[t])){
    t++;
    count++;
}
while ((u<wordsLen)&&(MTRX(puzzle,i-u,j+u)==LIST(words,n)[u])){
    u++;
    count++;
}
}
if (m==wordsLen){
    for (int y = 0; y < ROW(puzzle); y++){
        for (int z = 0; z< COL(puzzle); z++){
            MTRX(result,y,z) = '=';
        }
    }

    int limit = j + wordsLen;
    for (j; j< limit ; j++){
        MTRX(result,i,j) = MTRX(puzzle,i,j);
    }

    for (int a = 0; a < ROW(puzzle); a++){

```

```

        for (int b = 0; b < COL(puzzle); b++){
            if(b == COL(puzzle) - 1){
                printf("%c\n", MTRX(result, a, b));
            }
            else{
                printf("%c ", MTRX(result, a, b));
            }
        }

        printf("\n\n");
    }
    else if (o==wordsLen){
        for (int y = 0; y < ROW(puzzle); y++){
            for (int z = 0; z < COL(puzzle); z++){
                MTRX(result,y,z) = '=';
            }
        }

        int limit = i + wordsLen;
        for (i; i < limit ; i++){
            MTRX(result,i,j) = MTRX(puzzle,i,j);
        }

        for (int a = 0; a < ROW(puzzle); a++){
            for (int b = 0; b < COL(puzzle); b++){
                if(b == COL(puzzle) - 1){
                    printf("%c\n", MTRX(result, a, b));
                }
                else{
                    printf("%c ", MTRX(result, a, b));
                }
            }
        }

        printf("\n\n");
    }
}

```

```

else if (p==wordsLen){
    for (int y = 0; y < ROW(puzzle); y++){
        for (int z = 0; z< COL(puzzle); z++){
            MTRX(result,y,z) = '=';
        }
    }

    int start = j - wordsLen + 1;
    for (start; start <= j ; start++){
        MTRX(result,i,start) = MTRX(puzzle,i,start);
    }

    for (int a = 0; a < ROW(puzzle); a ++){
        for (int b = 0; b< COL(puzzle); b++){
            if(b == COL(puzzle) - 1){
                printf("%c\n", MTRX(result, a, b));
            }
            else{
                printf("%c ", MTRX(result, a, b));
            }
        }
    }
    printf("\n\n");
}
else if (q==wordsLen){
    for (int y = 0; y < ROW(puzzle); y++){
        for (int z = 0; z< COL(puzzle); z++){
            MTRX(result,y,z) = '=';
        }
    }

    int start = i - wordsLen + 1;
    for (start; start <= i ; start++){
        MTRX(result,start,j) = MTRX(puzzle,start,j);
    }
}

```



```

        for (int a = 0; a < ROW(puzzle); a++){
            for (int b = 0; b < COL(puzzle); b++){
                if(b == COL(puzzle) - 1){
                    printf("%c\n", MTRX(result, a, b));
                }
                else{
                    printf("%c ", MTRX(result, a, b));
                }
            }
            printf("\n\n");
        }
    else if (r==wordsLen){
        for (int y = 0; y < ROW(puzzle); y++){
            for (int z = 0; z < COL(puzzle); z++){
                MTRX(result,y,z) = '=';
            }
        }
        int limit= j + wordsLen;
        int curr = i;
        for (j; j< limit; j++){
            MTRX(result,curr,j) = MTRX(puzzle,curr,j);
            curr++;
        }

        for (int a = 0; a < ROW(puzzle); a++){
            for (int b = 0; b < COL(puzzle); b++){
                if(b == COL(puzzle) - 1){
                    printf("%c\n", MTRX(result, a, b));
                }
                else{
                    printf("%c ", MTRX(result, a, b));
                }
            }
        }
    }
}

```

```

        printf("\n\n");
    }
    else if (s==wordsLen){
        for (int y = 0; y < ROW(puzzle); y ++){
            for (int z = 0; z< COL(puzzle); z++){
                MTRX(result,y,z) = '=';
            }
        }
        int start = j - wordsLen + 1;
        int curr = i + wordsLen - 1;
        for (start; start<= j; start++){
            MTRX(result,curr,start) = MTRX(puzzle,curr,start);
            curr--;
        }

        for (int a = 0; a < ROW(puzzle); a ++){
            for (int b = 0; b< COL(puzzle); b++){
                if(b == COL(puzzle) - 1){
                    printf("%c\n", MTRX(result, a, b));
                }
                else{
                    printf("%c ", MTRX(result, a, b));
                }
            }
        }

        printf("\n\n");
    }
    else if (t==wordsLen){
        for (int y = 0; y < ROW(puzzle); y ++){
            for (int z = 0; z< COL(puzzle); z++){
                MTRX(result,y,z) = '=';
            }
        }
        int start = i - wordsLen + 1;
        int curr = j - wordsLen + 1;

```

```

        for (start; start<= i; start++){
            MTRX(result,start,curr) = MTRX(puzzle,start,curr);
            curr++;
        }

        for (int a = 0; a < ROW(puzzle); a ++){
            for (int b = 0; b< COL(puzzle); b++){
                if(b == COL(puzzle) - 1){
                    printf("%c\n", MTRX(result, a, b));
                }
                else{
                    printf("%c ", MTRX(result, a, b));
                }
            }
        }

        printf("\n\n");
    }
    else if (u==wordsLen){
        for (int y = 0; y < ROW(puzzle); y ++){
            for (int z = 0; z< COL(puzzle); z++){
                MTRX(result,y,z) = '=';
            }
        }
        int start = i - wordsLen + 1;
        int curr = j + wordsLen - 1;
        for (start; start<= i; start++){
            MTRX(result,start,curr) = MTRX(puzzle,start,curr);
            curr--;
        }

        for (int a = 0; a < ROW(puzzle); a ++){
            for (int b = 0; b< COL(puzzle); b++){
                if(b == COL(puzzle) - 1){
                    printf("%c\n", MTRX(result, a, b));
                }
            }
        }
    }
}

```

```

    }
    else{
        printf("%c ", MTRX(result, a, b));
    }
}

printf("\n\n");
}

}

}

clock_t endpuzzle = clock();

double duration = ((double) (endpuzzle - beginpuzzle))/(1000);

printf("Waktu dari eksekusi program adalah : %.5f sekon\n", (float)duration);
printf("Total perbandingan huruf yang dilakukan adalah sebanyak : %d kali", count);
}

```

Input dan Output (testcase)

a) Small – 1 (14 x 12)

AVENGER
MARVEL
BLACKWIDOW
HULK

LEVRAM = = = =

		W							
		O							
		D							
		I							
		W							
		K							
		C							
		A							
		L							
		B							

Waktu dari eksekusi program adalah : 0.05200 sekon
Total perbandingan huruf yang dilakukan adalah sebanyak : 809 kali

b) Small-2 (15 x 18)

INDONESIA
BEKASI
JAKARTA
JAMBI
MEDAN

===== ISAKEB =====

c)

Waktu dari eksekusi program adalah : 0.09500 sekon
Total perbandingan huruf yang dilakukan adalah sebanyak : 1567 kali

Waktu dari eksekusi program adalah : 0.09500 sekon
Total perbandingan huruf yang dilakukan adalah sebanyak : 1567 kali

A
L
I
U
Q
A

A
S
L
A
S
H
E
R

d) Medium-1 (20 x 22)

V
I
P
E
R
W
A
L
L

e) Medium-2 (24 x 20)

ARIES

[illegible]

f) Medium-3 (22 x 22)

NORWAY

ARRODNA

Waktu dari eksekusi program adalah : 0.13500 sekon
Total perbandingan huruf yang dilakukan adalah sebanyak : 3454 kali

g) Large-1 (32 x 30)

LZBEYDAJNYSPEAAEFNHHEEIFSEBSNCOR
LVBWBJCZLRSCQCLNMAOSSVFCQXNNHC
RIGILIKENTUMSRNNBMYRYUEQOHNRPYR
KTZAGSYTECCFUIETIEYACHUQANFRNCI
XUQTHIPIXGUXZAXLHFAHBMONWUHQOYCD
WLQJYQLTISDYIARFLFPFARZQVPITPO
HGXYGETRFRVFGVGMGADELWCXPFZVYU
KPAETQTOKQOFANRRFNGTATQNDQKMCLA
SJOENAVBZCDQTOHTKYOLCWEZGAUJAP
EZBCLBNMFHQHMLWLRONMSYIAJ3BNYEY
JHDLGNYFOGTHBKTROVIOUDRBEXLSJ
GXEYBDCUJYJTXZJABMARHXCFUZZLUZL
BBSRWBTBUBOORANDFEXTWYDYSVAHDK
SERATANAEELNATHMKIGYMAEYKJLNYLZW
KYSGUUMJCCSDXYHMGSONRSYNNPPYIS
MVEQBWFFFUGTVLEGIROFENWNEREXVUXS
UMRLPHYRBAZECSMCRUSIGEQQOKUMTVH
VLPGQDXFAUULLGYJYTRAMAFANULCPGS
SAMYDQXHAKNAPTQJTNCTULEYYSLZNWU
UURTQFXLXHXHDLZFUXILUADQB3OPERM
ICQCPWDSHTADARDBQNYPSADQVPTJUN
RSRCEQPOVZNHGGSQTRHSXALUALAHSG
IYBGBUAFVWHTAGTGAJMRGRPEXTHVVSNL
SQSAZARZVPMNQSMRSLNXPWDDAPDXHW
CHRJLIZUTYQJQLSRMFPFAVLDRILRUZ
NWNWPLRDSNNCCRMKEODDKKJFVLRJBX
NAUUFYUWQLCSWDABHCPREJYPSLBAR
UILDUMVXSVVQIZPAAZWKJYQXXHADEN
RDHUKRTSFUBMYRQRKKDZYRGUMMUUYG
KMFHGFZWZQFWZFWAHGEPOXPXTODHGG
MVEQBWFFFUGTVLEGIROFENWNEREXVUXS
UMRLPHYRBAZECSMCRUSIGEQQOKUMTVH

A 20x20 grid of dots. The letters N, O, Y, C, O, R, P are placed in the grid such that they read 'NOY CORP' from top to bottom, left to right. The letters are arranged in a staggered pattern: N is at (1,12), O is at (2,13), Y is at (3,14), C is at (4,15), O is at (5,16), R is at (6,17), and P is at (7,18). The rest of the grid is empty.

Waktu dari eksekusi program adalah : 0.26500 sekon
Total perbandingan huruf yang dilakukan adalah sebanyak : 5180 kali

ALU AH5

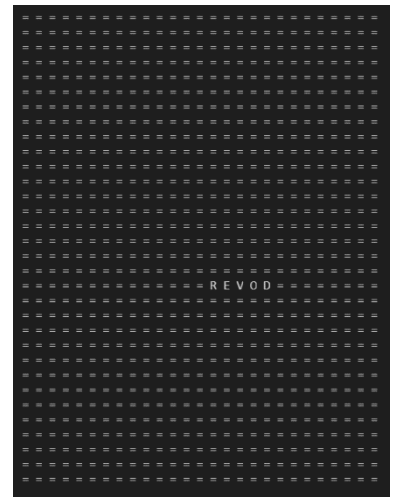
A
C
T
P
S

h) Large-2 (32 x 30)

FESSAALLACLIATMCNOSTSOBODUQHAB
 IWTBWLHOLLINLANSTNGGLEBADA
 LRLCUDVYTPNFRANKFORTURDPRGHI
 SAOBPDYFCRVRVCMKSTSELRAHCTJKL
 QLPTQXDVONHWEZSIOBDELHVVFMMO
 TEADJLKNNOJVPYRYNIBBJXTOPQR
 IINJWOCWCSWPPPTTOPAADCQIRST
 GGAECPOXOIQIUHIHINTHEZZJDVWX
 VHIFCRRTDRGRCNCNVOEENAWIAJJKL
 UDDFOKEJDANEQAFNDELNGYEIJKL
 QLNELRLIGKODMQRTAEFEETLBJKL
 SEIRUFTALBUMOHMTLWNYCILMJKL
 AITSMTJLEDZUHNSSAAEVCUJJKL
 CFQOBWITUEPGXACMLQHNRVLJJKL
 RGDNUMLDCNETLLRIGRPTCYHOHJJKL
 ANSCSAONEAEATNCLKRMRESHSSCJJKL
 MIFISKSESTSAAHOURCIVYURAVJKL
 ERGTLDCSKPMEUBMTOLQVUANCJJKL
 NPNYTOURHCFOSWREVODWMCXOJKL
 TSUVUGPOAAAIIETLWPTNIAJJKL
 OKOKUKEATMRJNNMONTGOMERYJKL
 NRPAQTNNRSTRDENVERHPQAUJJKL
 PGMKIZAFANOIJZESXPAKEPOTJJKL
 RXXXCSLHGNAYBQMELASFGWVJJKL
 GGAECPOXOIQIUHIHINTHEZZJDJJKL
 VHIFCRRTDRGRCNCNVOEENAWIAJJKL
 UDDFOKEJDANEQAFNDELNGYEIJKL
 QLNELRLIGKODMQRTAEFEETLBJKL
 SEIRUFTALBUMOHMTLWNYCILMJKL
 AITSMTJLEDZUHNSSAAEVCUJJKL
 VHIFCRRTDRGRCNCNVOEENAWIAJJKL
 UDDFOKEJDANEQAFNDELNGYEIJKL

ALBANY
 DAVEN
 INDIANAPOLIS
 RICHMOND

The diagram shows a 2D hexagonal lattice. A central vertex is labeled 'Y'. To its right is a vertex labeled 'N'. Above 'N' is a vertex labeled 'A'. Above 'A' is a vertex labeled 'B'. Above 'B' is a vertex labeled 'L'. Above 'L' is a vertex labeled 'A'. The lattice extends to the left and right, with the right side being truncated.



REVOD

[illegible]

Waktu dari eksekusi program adalah : 0.23800 sekon
Total perbandingan huruf yang dilakukan adalah sebanyak : 5102 kali

i) Large-3 (32 x 30)

EESSAWLLLATHMCNOTSOBODUQHABE
 TWIBNLLOCLTNLAWSNINGGLEBEAFED
 LRLCLVVTNPNFRANKFORTURDPGRHT
 SAOBPDVYFCRVWMCNOTSLELRAHCTJKL
 QLPITQXDVTNMZESIOBOELHVVFMNO
 TEADJLKKNNQOVPVRYRNVIBBJXTOPQ
 ITHJWOCWESWPPTTOPOAADQCIRSTU
 VGAEFCPOXOITQIUHTHINTHEZZJVDVW
 GHIFCRRTRDGRCCNYOEENAWIAXYZ
 UDDFOKEJDANEQAFNDELNGYEIJJKL
 QLNELRLRIGMKDMRMTAJEEETLBJKL
 SEIRUFTTALABUMOTNLWNYCILMJJKL
 AITSMATTLJEDZUHNNSAAEVCUIJJKL
 CFQOBBWITUEPGTACBMIOHRNVULJKL
 RGDUNLNDCEILLRIGPTCYOHOJJKL
 ANSCSAONEAEATNKLRRMESHSSCJJKL
 MTFISKEKTSAAHOURCYIVURAVJJKL
 ERGTLDCSTPMEUBSMTOLDQVWANCJKL
 NPNYIUORHSFOSWREVOVMCOXJJKL
 TSUVUWGPAAEIIETLUAPTNTASJJKL
 OKOKUKEATERNNNMONTGOMERYJKL
 ARPAQNTNNRSTODENVERHPQAUJJKL
 IGMKIZAFANOJIDEXSPHAKEPOTJJKL
 LRRXCSPHGNAYBQNELASFZWZTJJKL
 AGAECPXOITQIUHINLTHEZZJDKL
 RHIFCRRTRDGRCCNYOEENAWIAJJKL
 TDDFOKEJDANEQAFNDELPANVOLJKL
 SLNELRLRIGMKDMRMTAJEEETLBJKL
 UEIRUFTTALABUMOTNLWNYCILMJJKL
 AITSMATTLJEDZUHNNSAAEVCUIJJKL
 VHIFCRRTRDGRCCNYOEENAWIAJJKL
 UDDFOKEJDANEQAFNDELNGYEIJJKL

A
I
L
A
R
T
S
U
A
A

Figure 1 shows a 2D grid of points. A central region is marked with 'A' and 'I' labels. The 'A' label is at the center, and the 'I' label is at the center of the 'A' region. The 'A' region is a 4x4 grid of points, and the 'I' region is a 4x4 grid of points. The 'A' region is the innermost, and the 'I' region is the outermost. The labels 'A' and 'I' are placed at the center of their respective regions.

= E S P A N Y O L

Total perbandingan huruf yang dilakukan adalah sebanyak : 5466 kali

BAB 4

Drive

Berikut file repository github untuk Tugas Kecil 1 Strategi Algoritma :

<https://github.com/yosalx/wordpuzzle>