

TUGAS KECIL IF2211 STRATEGI ALGORITMA

Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force



Semester II Tahun 2022/2023

Disusun Oleh:

13521123

William Nixon

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

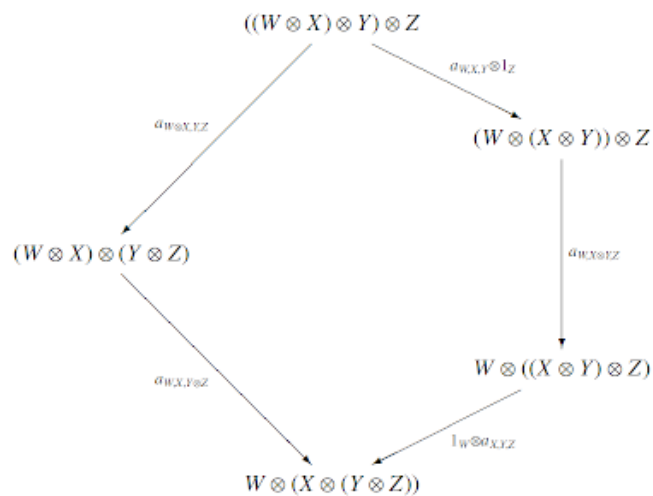
Bagian I

Deskripsi Algoritma

1. Algoritma Pemecahan Puzzle Kartu 24

Untuk mencari seluruh solusi jika diberikan 4 buah kartu dengan angka tertentu, maka akan dicoba seluruh kombinasi urutan kartu berikut dengan operator-operator yang dapat digunakan, yaitu operator tambah (+), kurang (-), bagi (/), kali (*) dan kurung. Adapun rangkaian algoritma yang digunakan untuk memecahkannya sebagai berikut:

1. Bangkitkan sebuah kemungkinan pengacakan kartu input dari user.
2. Untuk setiap kemungkinan urutan kartu K1 K2 K3 K4, kombinasikan dengan operator yang tersedia. Hasil kombinasi akan menyerupai K1 op1 K2 op2 K3 op3 K4 di mana op merupakan salah satu dari 4 jenis operator legal yaitu +, -, *, dan /.
3. Setelah urutan kartu dan operator ditentukan, maka uji setiap kemungkinan dibawah 5 kemungkinan pengurungan yang berbeda. Pengurungan akan berpengaruh terhadap urutan evaluasi dari ekspresi. Pada kasus ini, jumlah cara menyusun kurung untuk 4 operan adalah 5 (angka Catalan ketiga)



Gambar 1.1. Jumlah Cara Pengurungan 4 Buah Term

4. Evaluasi hasil dari ekspresi yang telah dibentuk. Jika hasil tersebut sama dengan 24, maka kombinasi tersebut merupakan salah satu solusi dari permainan. Masukkan ke dalam sebuah set untuk di output nantinya.
5. Ulangi langkah 1-4 sampai seluruh kemungkinan pengacakan kartu telah dihasilkan.

2. Algoritma Pengacakan Kartu

Algoritma pengacakan tiap kartu dilakukan secara sederhana dengan metode brute-force menggunakan 4 buah nested for-loop. Indeks for loop pertama menentukan indeks dari kartu pertama, indeks for loop kedua menentukan indeks kartu kedua, for loop ketiga menentukan indeks dari kartu ketiga, dan for loop keempat menentukan indeks dari kartu keempat. Pada setiap for loop, dilakukan pengujian bahwa indeks dari masing-masing for loop tersebut tidak ada yang sama untuk mencegah *double-counting*.

Bagian II Source Code

Program pada mulanya dibuat di dalam 1 file dengan berbagai fungsi pembantu yang memiliki tanggung jawab tersendiri. Karena penulis merasa program masih berskala kecil, penulis memutuskan untuk tidak melakukan refactoring dan memecah fungsi-fungsi main.cpp ke dalam modul-modul lain. Berikut merupakan kode program yang disimpan di file main.cpp tersebut.

```
#include <iostream>
#include <sstream>
#include <fstream>
#include <string>
#include <random>
#include <string>
#include <stack>
#include <set>
#include <chrono>

using namespace std;

// Global variables
int DIGITS[4];
const int MAX_VAL = 13;
const char OPS[] = {'+', '-', '*', '/'};
set<string> result;

// Function prototypes
int displayMode();
void populateDigits(int mode);
void displayDigits();
void solve();
int convert(string input);
void processDigits(int num1, int num2, int num3, int num4);
void displayResult(int count);
void handleSave(float count);
```

```

int main()
{
    cout << "----- Selamat datang di permainan 24! -----" << endl;

    int mode = displayMode();
    populateDigits(mode);
    displayDigits();

    auto t1 = chrono::high_resolution_clock::now();

    solve();

    auto t2 = chrono::high_resolution_clock::now();
    auto ms_int = std::chrono::duration_cast<std::chrono::milliseconds>(t2 - t1);

    displayResult(ms_int.count());
    handleSave(ms_int.count());
}

// Get pengaturan pemain, ulangi pilihan tersebut valid.
int displayMode()
{
    char choice = -1;
    do
    {
        cout << "======" << endl;
        cout << "Pilih mode permainan:" << endl;
        cout << "1: Random" << endl;
        cout << "2: Input" << endl;
        cout << "======" << endl;
        cin >> choice;
    } while (choice != '1' && choice != '2');
    return int(choice) - '0';
}

// Isi array digits dengan angka random / dengan input pengguna.
void populateDigits(int mode)
{
    switch (mode)
    {
        case 1:
            // Seed dengan time saat ini.
            srand(time(0));
            DIGITS[0] = rand() % MAX_VAL + 1;
            DIGITS[1] = rand() % MAX_VAL + 1;
            DIGITS[2] = rand() % MAX_VAL + 1;
            DIGITS[3] = rand() % MAX_VAL + 1;
            break;
        case 2:
            int curr_idx = 0;
            // Loop validasi input
            do

```

```

    {
        string temp;
        cout << "Digit " << curr_idx + 1 << " : ";
        cin >> temp;
        int converted = convert(temp);
        if (converted >= 1 && converted <= 13)
        {
            DIGITS[curr_idx] = converted;
            curr_idx += 1;
        }
        else
        {
            cout << "Input invalid!" << endl;
        }
    } while (curr_idx != 4);
    break;
}
}

```

```

// Tampilkan digit yang telah diterima program.
void displayDigits() {
    cout << "Angka-angka anda adalah : " << endl;
    for (auto x : DIGITS)
    {
        cout << x << " ";
    }
    cout << endl << endl;
}

```

```

// Berusaha ubah input digit pemain menjadi angka
int convert(string input)
{
    try
    {
        if (input == "A" || input == "a")
        {
            return 1;
        }
        else if (input == "J" || input == "j")
        {
            return 11;
        }
        else if (input == "Q" || input == "q")
        {
            return 12;
        }
        else if (input == "K" || input == "k")
        {
            return 13;
        }
        return stoi(input);
    }
}

```

```

        catch (...)
        {
            return -1;
        }
    }

// Pengaturan operasi antara 2 buah angka dan 1 karakter operator
float operation(float num1, float num2, char op) {
    switch (op) {
        case '+':
            return num1 + num2;
        case '-':
            return num1 - num2;
        case '/':
            return num1 / num2;
        case '*':
            return num1 * num2;
    }
}

// Tampilkan hasil eksekusi program dan juga solusi.
void displayResult(int ms_int) {
    if (result.size() == 0) {
        cout << "Tidak terdapat solusi" << endl;
    } else {
        cout << "Terdapat " << result.size() << " solusi." << endl;
        for (auto &str : result)
        {
            cout << str << endl;
        }
        cout << endl;
    }

    cout << "Waktu yang dibutuhkan " << ms_int << " ms" << endl;
}

//Coba segala permutasi dari 4 digit yang telah diterima program.
void solve() {
    for (int digit1 = 0; digit1 < 4; digit1++)
    {
        for (int digit2 = 0; digit2 < 4; digit2++)
        {
            for (int digit3 = 0; digit3 < 4; digit3++)
            {
                for (int digit4 = 0; digit4 < 4; digit4++)
                {
                    if (digit1 == digit2 || digit1 == digit3 || digit1 == digit4 ||
digit2 == digit3 || digit2 == digit4 || digit3 == digit4)
                    {
                        continue;
                    }
                    processDigits(DIGITS[digit1], DIGITS[digit2], DIGITS[digit3],

```

```

DIGITS[digit4]);
    }
}
}
}

// Proses semua kemungkinan jika diberikan 4 angka berurutan.
// Coba segala kombinasi dari operasi, dan juga coba segala kombinasi kurung.
void processDigits(int num1, int num2, int num3, int num4)
{
    for (int op1 = 0; op1 < 4; op1++)
    {
        for (int op2 = 0; op2 < 4; op2++)
        {
            for (int op3 = 0; op3 < 4; op3++)
            {
                ostringstream stream;
                string new_string = stream.str();
                float firstPair = operation(num1, num2, OPS[op1]);
                float secondPair = operation(num2, num3, OPS[op2]);
                float thirdPair = operation(num3, num4, OPS[op3]);

                // (NUM OP NUM) OP (NUM OP NUM)
                float firstPossibility = operation(firstPair, thirdPair, OPS[op2]);

                if (abs(firstPossibility-24) < 0.0001) {
                    stream << "( " << num1 << " " << OPS[op1] << " " << num2 << " "
" << OPS[op2] << " ( " << num3 << " " << OPS[op3] << " " << num4 << " )";
                    new_string = stream.str();
                    result.insert(new_string);
                }

                // ((NUM OP NUM) OP NUM) OP NUM
                float secondPossibility = operation(operation(firstPair, num3,
OPS[op2]), num4, OPS[op3]);

                if (abs(secondPossibility - 24) < 0.0001)
                {
                    stream.str("");
                    stream.clear();
                    stream << "( ( " << num1 << " " << OPS[op1] << " " << num2 << "
) " << OPS[op2] << " " << num3 << " ) " << OPS[op3] << " " << num4;
                    new_string = stream.str();
                    result.insert(new_string);
                }

                // (NUM op (NUM op NUM)) op NUM
                float thirdPossibility = operation(operation(num1, secondPair,
OPS[op1]), num4, OPS[op3]);

                if (abs(thirdPossibility - 24) < 0.0001)
                {

```

```

        stream.str("");
        stream.clear();
        stream << "( " << num1 << " " << OPS[op1] << " ( " << num2 << "
" << OPS[op2] << " " << num3 << " ) ) " << OPS[op3] << " " << num4;
        new_string = stream.str();
        result.insert(new_string);
    }

    float fourthPossibility = operation(num1, operation(secondPair,
num4, OPS[op3]), OPS[op1]);

    // NUM op ((NUM op NUM) op NUM)
    if (abs( fourthPossibility - 24) < 0.0001)
    {
        stream.str("");
        stream.clear();
        stream << num1 << " " << OPS[op1] << " ( ( " << num2 << " " <<
OPS[op2] << " " << num3 << " ) " << OPS[op3] << " " << num4 << " ) ";
        new_string = stream.str();
        result.insert(new_string);
    }

    float fifthPossibility = operation(num1, operation(num2, thirdPair,
OPS[op2]),OPS[op1]);

    // NUM op (NUM op (NUM op NUM))
    if (abs(fifthPossibility - 24) < 0.0001)
    {
        stream.str("");
        stream.clear();
        stream << num1 << " " << OPS[op1] << " ( " << num2 << " " <<
OPS[op2] << " ( " << num3 << " " << OPS[op3] << " " << num4 << " ) )";
        new_string = stream.str();
        result.insert(new_string);
    }
    }
}

// Handle proses penulisan hasil ke dalam file
void handleSave(float time)
{
    cout << "Apakah ingin menyimpan solusi? (Y/T)" << endl;

    string save;
    do
    {
        cin >> save;
    } while (save != "Y" && save != "T" && save != "y" && save != "t");

    if (save == "Y" || save == "y")
    {

```



```

string filename;
cout << "Masukkan nama file: ";
do
{
    cin >> filename;
} while (filename.empty());

string filedir = "test/";
filedir.append(filename);
ofstream MyFile(filedir);

if (result.size() == 0) {
    MyFile << "Tidak terdapat solusi" << endl;
} else {
    MyFile << "Terdapat " << result.size() << " solusi." << endl;
    for (auto &str : result)
    {
        MyFile << str << endl;
    }
    MyFile << endl;
}

MyFile << "Waktu yang dibutuhkan " << time << " ms" << endl;
MyFile.close();
}
}

```

Bagian III

Hasil Uji

Diuji dengan ASUS Vivobook dengan spesifikasi:

1. RAM 4 GB
2. Prosesor Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz 2.30 GHz

No	Masukan	Keluaran
1	5 6 6 13 (rand)	<pre> ^ PS C:\Users\ASUS\Desktop\tucil> ./bin/main ----- Selamat datang di permainan 24! ----- ===== Pilih mode permainan: 1: Random 2: Input ===== 1 Angka-angka anda adalah : 5 6 6 13 Tidak terdapat solusi Waktu yang dibutuhkan 2 ms Apakah ingin menyimpan solusi? (Y/T) y Masukkan nama file: tes.txt</pre>

2

1 2 3 4 (dipotong)

```
PS C:\Users\ASUS\Desktop\tucil> ./bin/main
----- Selamat datang di permainan 24! ---
=====
Pilih mode permainan:
1: Random
2: Input
=====
2
Digit 1 : 1
Digit 2 : 2
Digit 3 : 3
Digit 4 : 4
Angka-angka anda adalah :
1 2 3 4

Terdapat 242 solusi.
( ( 1 * 2 ) * 3 ) * 4
( ( 1 * 2 ) * 4 ) * 3
( ( 1 * 3 ) * 2 ) * 4
( ( 1 * 3 ) * 4 ) * 2
( ( 1 * 4 ) * 2 ) * 3
( ( 1 * 4 ) * 3 ) * 2
( ( 1 + 2 ) + 3 ) * 4
( ( 1 + 3 ) + 2 ) * 4
( ( 2 * 1 ) * 3 ) * 4
( ( 2 * 1 ) * 4 ) * 3
( ( 2 * 3 ) * 1 ) * 4
( ( 2 * 3 ) * 4 ) * 1
( ( 2 * 3 ) * 4 ) / 1
( ( 2 * 3 ) / 1 ) * 4
( ( 2 * 4 ) * 1 ) * 3
( ( 2 * 4 ) * 3 ) * 1
( ( 2 * 4 ) * 3 ) / 1
( ( 2 / 1 ) * 3 ) * 4
4 / ( 1 / ( 2 * 3 ) )
4 / ( 1 / ( 3 * 2 ) )

Waktu yang dibutuhkan 3 ms
Apakah ingin menyimpan solusi? (Y/T)
y
Masukkan nama file: 1 2 3 4.txt
```

3

AJQK

```
=====
2
Digit 1 : A
Digit 2 : J
Digit 3 : Q
Digit 4 : K
Angka-angka anda adalah :
1 11 12 13

Terdapat 32 solusi.
( ( 1 * 13 ) - 11 ) * 12
( ( 13 * 1 ) - 11 ) * 12
( ( 13 - 11 ) * 1 ) * 12
( ( 13 - 11 ) * 12 ) * 1
( ( 13 - 11 ) * 12 ) / 1
( ( 13 - 11 ) / 1 ) * 12
( ( 13 / 1 ) - 11 ) * 12
( 1 * ( 13 - 11 ) ) * 12
( 1 * 12 ) * ( 13 - 11 )
( 12 * ( 13 - 11 ) ) * 1
( 12 * ( 13 - 11 ) ) / 1
( 12 * 1 ) * ( 13 - 11 )
( 12 / 1 ) * ( 13 - 11 )
( 13 - ( 1 * 11 ) ) * 12
( 13 - ( 11 * 1 ) ) * 12
( 13 - ( 11 / 1 ) ) * 12
( 13 - 11 ) * ( 1 * 12 )
( 13 - 11 ) * ( 12 * 1 )
( 13 - 11 ) * ( 12 / 1 )
( 13 - 11 ) / ( 1 / 12 )
1 * ( ( 13 - 11 ) * 12 )
1 * ( 12 * ( 13 - 11 ) )
12 * ( ( 1 * 13 ) - 11 )
12 * ( ( 13 * 1 ) - 11 )
12 * ( ( 13 - 11 ) * 1 )
12 * ( ( 13 - 11 ) / 1 )
12 * ( ( 13 / 1 ) - 11 )
12 * ( 1 * ( 13 - 11 ) )
12 * ( 13 - ( 1 * 11 ) )
12 * ( 13 - ( 11 * 1 ) )
12 * ( 13 - ( 11 / 1 ) )
12 / ( 1 / ( 13 - 11 ) )

Waktu yang dibutuhkan 1 ms
Apakah ingin menyimpan solusi? (Y/T)
y
Masukkan nama file: A_J_Q_K.txt
```

4	J Q K J	<pre> 2 Digit 1 : J Digit 2 : Q Digit 3 : K Digit 4 : J Angka-angka anda adalah : 11 12 13 11 Terdapat 32 solusi. ((12 - 11) * 11) + 13 ((12 - 11) * 13) + 11 (11 * (12 - 11)) + 13 (11 + 13) * (12 - 11) (11 + 13) / (12 - 11) (11 / (12 - 11)) + 13 (12 + 13) - (11 / 11) (12 - (11 / 11)) + 13 (12 - 11) * (11 + 13) (12 - 11) * (13 + 11) (13 * (12 - 11)) + 11 (13 + 11) * (12 - 11) (13 + 11) / (12 - 11) (13 + 12) - (11 / 11) (13 - (11 / 11)) + 12 (13 / (12 - 11)) + 11 11 + ((12 - 11) * 13) 11 + (13 * (12 - 11)) 11 + (13 / (12 - 11)) 11 - ((11 - 12) * 13) 11 - (13 * (11 - 12)) 11 - (13 / (11 - 12)) 12 + (13 - (11 / 11)) 12 - ((11 / 11) - 13) 13 + ((12 - 11) * 11) 13 + (11 * (12 - 11)) 13 + (11 / (12 - 11)) 13 + (12 - (11 / 11)) 13 - ((11 - 12) * 11) 13 - ((11 / 11) - 12) 13 - (11 * (11 - 12)) 13 - (11 / (11 - 12)) Waktu yang dibutuhkan 0 ms Apakah ingin menyimpan solusi? (Y/T) y Masukkan nama file: J Q K J.txt </pre>
5	Invalid input, kemudian a a a a	<pre> ===== Pilih mode permainan: 1: Random 2: Input ===== 2 Digit 1 : a Digit 2 : b Input invalid! Digit 2 : c Input invalid! Digit 2 : d Input invalid! Digit 2 : a Digit 3 : a Digit 4 : a Angka-angka anda adalah : 1 1 1 1 Tidak terdapat solusi Waktu yang dibutuhkan 1 ms Apakah ingin menyimpan solusi? (Y/T) </pre>

6

A 8 9 Q (input case insensitive)

```

2
Digit 1 : A
Digit 2 : 8
Digit 3 : 9
Digit 4 : q
Angka-angka anda adalah
1 8 9 12

```

```

Terdapat 49 solusi.
( ( 1 * 12 ) - 9 ) * 8
( ( 1 + 9 ) - 8 ) * 12
( ( 1 - 8 ) + 9 ) * 12
( ( 12 * 1 ) - 9 ) * 8
( ( 12 - 9 ) * 1 ) * 8
( ( 12 - 9 ) * 8 ) * 1
( ( 12 - 9 ) * 8 ) / 1
( ( 12 - 9 ) / 1 ) * 8
( ( 12 / 1 ) - 9 ) * 8
( ( 9 + 1 ) - 8 ) * 12
( ( 9 - 8 ) + 1 ) * 12
( 1 * ( 12 - 9 ) ) * 8
( 1 * 8 ) * ( 12 - 9 )
( 1 + ( 9 - 8 ) ) * 12
( 1 - ( 8 - 9 ) ) * 12
( 12 - ( 1 * 9 ) ) * 8
( 12 - ( 9 * 1 ) ) * 8
( 12 - ( 9 / 1 ) ) * 8
( 12 - 9 ) * ( 1 * 8 )
( 12 - 9 ) * ( 8 * 1 )
( 12 - 9 ) * ( 8 / 1 )
( 12 - 9 ) / ( 1 / 8 )
( 8 * ( 12 - 9 ) ) * 1
( 8 * ( 12 - 9 ) ) / 1
( 8 * 1 ) * ( 12 - 9 )
( 8 / 1 ) * ( 12 - 9 )
( 9 + ( 1 - 8 ) ) * 12
( 9 - ( 8 - 1 ) ) * 12
1 * ( ( 12 - 9 ) * 8 )
1 * ( 8 * ( 12 - 9 ) )
12 * ( ( 1 + 9 ) - 8 )
12 * ( ( 1 - 8 ) + 9 )
12 * ( ( 9 + 1 ) - 8 )
12 * ( ( 9 - 8 ) + 1 )
12 * ( 1 + ( 9 - 8 ) )
12 * ( 1 - ( 8 - 9 ) )
12 * ( 9 + ( 1 - 8 ) )

```

```

12 * ( 9 + ( 1 - 8 ) )
12 * ( 9 - ( 8 - 1 ) )
8 * ( ( 1 * 12 ) - 9 )
8 * ( ( 12 * 1 ) - 9 )
8 * ( ( 12 - 9 ) * 1 )
8 * ( ( 12 - 9 ) / 1 )
8 * ( ( 12 / 1 ) - 9 )
8 * ( 1 * ( 12 - 9 ) )
8 * ( 12 - ( 1 * 9 ) )
8 * ( 12 - ( 9 * 1 ) )
8 * ( 12 - ( 9 / 1 ) )
8 / ( ( 12 / 9 ) - 1 )
8 / ( 1 / ( 12 - 9 ) )

```

```

Waktu yang dibutuhkan 0 ms
Apakah ingin menyimpan solusi? (Y/T)
y
Masukkan nama file: a_8_9_q.txt

```

7	2 3 5 12 (terdapat pembagian dan pengurangan pecahan)	<pre> t PS C:\Users\ASUS\Desktop\tucil> ./bin/ ----- Selamat datang di permainan 24! ===== Pilih mode permainan: 1: Random 2: Input ===== 2 Digit 1 : 2 Digit 2 : 3 Digit 3 : 5 Digit 4 : 12 Angka-angka anda adalah : 2 3 5 12 Terdapat 1 solusi. 12 / (3 - (5 / 2)) Waktu yang dibutuhkan 1 ms Apakah ingin menyimpan solusi? (Y/T) </pre>
8	1 3 4 6 (terdapat pembagian dan pengurangan pecahan)	<pre> ----- Selamat datang di permainan 24! ===== Pilih mode permainan: 1: Random 2: Input ===== 2 Digit 1 : 1 Digit 2 : 3 Digit 3 : 4 Digit 4 : 6 Angka-angka anda adalah : 1 3 4 6 Terdapat 1 solusi. 6 / (1 - (3 / 4)) Waktu yang dibutuhkan 1 ms Apakah ingin menyimpan solusi? (Y/T) </pre>
9	3 5 7 13 (terdapat pembagian hasil akhir)	<pre> ----- Selamat datang di permainan 24! ----- ===== Pilih mode permainan: 1: Random 2: Input ===== 2 Digit 1 : 3 Digit 2 : 5 Digit 3 : 7 Digit 4 : 13 Angka-angka anda adalah : 3 5 7 13 Terdapat 4 solusi. ((13 * 5) + 7) / 3 ((5 * 13) + 7) / 3 (7 + (13 * 5)) / 3 (7 + (5 * 13)) / 3 Waktu yang dibutuhkan 1 ms Apakah ingin menyimpan solusi? (Y/T) </pre>

10	1 6 6 8 (terdapat pengurangan pecahan)	<pre> PS C:\Users\yasus\Desktop\tucil> ./bin/main ----- Selamat datang di permainan 24! ----- ===== Pilih mode permainan: 1: Random 2: Input ===== 2 Digit 1 : 1 Digit 2 : 6 Digit 3 : 6 Digit 4 : 8 Angka-angka anda adalah : 1 6 6 8 Terdapat 1 solusi. 6 / (1 - (6 / 8)) Waktu yang dibutuhkan 1 ms Apakah ingin menyimpan solusi? (Y/T) </pre>
----	--	---

Bagian IV Lain-Lain

1. Link Repository

https://github.com/williamnixon20/tucil_stima_1_13521123

2. Checklist Penilaian

Poin	Ya	Tidak
Program berhasil dikompilasi tanpa kesalahan	Ya	
Program berhasil running	Ya	
Program dapat membaca input / generate sendiri dan memberikan luaran	Ya	
Solusi yang diberikan program memenuhi (berhasil mencapai 24)	Ya	
Program dapat menyimpan solusi dalam file teks	Ya	