

Tugas Kecil 1 IF2211 Strategi Algoritma
Semester II tahun 2022/2023

Penyelesaian Permainan Kartu 24 dengan Algoritma *Brute Force*

Disusun oleh :

Muhamad Salman Hakim Alfarisi (13521010)



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2023**

BAB I

Algoritma Brute Force

Strategi brute force merupakan suatu strategi algoritma untuk menyelesaikan suatu persoalan dengan cara mencoba semua kemungkinan solusi. Algoritma ini umumnya tidak mangkus dikarenakan memerlukan volume komputasi yang besar dalam penyelesaian suatu persoalan. Oleh karena itu, algoritma ini sering dijadikan pembanding dengan algoritma lain. Meskipun dinilai kurang mangkus, algoritma brute force memiliki kelebihan dapat menyelesaikan hampir semua permasalahan yang tidak dapat diselesaikan dengan algoritma lain.

Salah satu penerapan algoritma brute force adalah pada permainan 24. Permainan 24 adalah permainan yang dimainkan menggunakan empat kartu remi yang dikeluarkan secara acak sehingga dapat menghasilkan angka 24 dari perhitungan nilai kartu. Tujuan permainan ini adalah melakukan perhitungan secara cepat menemukan angka 24 dari kombinasi perhitungan kali (\times), bagi ($/$), tambah ($+$), dan kurang ($-$). Kartu bernilai sesuai angka atau huruf yang terdapat pada kartu remi. Terdapat 52 kartu yang terdiri dari 4 suit (hati, wajik, sekop, dan keriting) yang masing-masing memiliki 13 kartu (As (A), 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack (J), Queen (Q), dan King (K)). Kartu As bernilai 1, kartu Jack bernilai 11, kartu Queen bernilai 12, dan kartu King bernilai 13. Kartu lain yang sudah dalam bentuk bilangan, merepresentasikan bilangan itu sendiri.

Proses algoritma brute force dalam penyelesaian masalah permainan 24 dilakukan dengan menerima empat kartu yang kemudian dibuat sebuah set. Keempat angka tersebut akan ditukar dengan semua kemungkinan posisi dengan melakukan perulangan. Setelah itu, algoritma ini akan mencoba semua kombinasi dari 4 operator matematika ($+$, $-$, \times , $/$) dengan melakukan perulangan. Selain itu, 4 angka dan 4 buah operator tersebut akan dievaluasikan dalam 5 jenis tipe kurung, yaitu :

1. Tipe 1 : (a operator1 b) operator2 (c operator3 d)
2. Tipe 2 : ((a operator1 b) operator2 c) operator3 d
3. Tipe 3 : (a operator1 (b operator2 c)) operator3 d
4. Tipe 4 : a operator1 ((b operator2 c) operator3 d)
5. Tipe 5 : a operator1 (b operator2 (c operator3 d))

Jika evaluasi dari kombinasi pertukaran angka, operasi, dan tipe kurung menghasilkan nilai 24, evaluasi tersebut akan disimpan dalam bentuk string ke dalam sebuah set. Set dirancang tidak akan menerima string evaluasi yang duplikat sehingga tidak akan menemukan solusi yang duplikat. Program akan menampilkan solusi dan runtime. Hasil solusi juga dapat disimpan dalam bentuk .txt.

BAB II

Source Code Program dalam C++

a. operations.hpp

```
#include <iostream>
#include <time.h> // random and runtime
#include <vector> // vector of string result
#include <fstream> // write to file
#include <iomanip> // setprecision of runtime

using namespace std;

#ifndef OPERATIONS_HPP
#define OPERATIONS_HPP

double convertInputs(string x);
/* I.S. Receive input string from card*/
/* F.S. Return card value in double type */

bool validInputs(string x);
/* I.S. Receive input string from card*/
/* F.S. Return true if input valid, false if not */

double zeroDiv(int a, int b);
/* I.S. Receive 2 integer*/
/* F.S. Return 0 if b = 0, else return a/b */

double calc(double a, double b, int op);
/* I.S. Receive 2 double of numbers and 1 integer of operation */
/* F.S. Return result of calculation */

bool checkDuplicate(vector<string> result, string temporary);
/* I.S. Receive 1 vector and 1 string*/
/* F.S. Return true if string is already in vector, false if not in the
vector */
```

```

void printResult(vector <string> result);
    /* I.S. Receive 1 vector*/
    /* F.S. Print all element in vector */

string exp(int sign);
    /* I.S. Receive 1 integer*/
    /* F.S. Return string of operator */

void splashScreen();
    /* I.S. None*/
    /* F.S. Print splash screen */

void totalSolutions(vector <string> result);
    /* I.S. Receive 1 vector*/
    /* F.S. Print total solutions */

void writeToFile(vector <string> result, string input1, string input2,
string input3, string input4, double runtime);
    /* I.S. Receive 1 vector, 4 string, and 1 double*/
    /* F.S. Write all element in vector to file */

#endif

```

b. operation.cpp

```

#include "operations.hpp"

void splashScreen() {
    cout << "-----" <<
endl;
    cout << "                Welcome to MAKE IT 24!                " <<
endl;
    cout << "This program will solve the Make It 24 game for you." <<
endl;
    cout << "                .-----." <<
endl;
    cout << "                .-----.      |A .      |      " <<
endl;
    cout << "                |A_ _ |      .-----; / \ \      |      " <<
endl;
    cout << "                |( \ \ / )|-----. _      |(_,_ ) |      " <<
endl;
    cout << "                | \ \ / | / \ \      |( )      | I A|      " <<
endl;

```

```

        cout << "          |  \\/ A|/  \\/ |_x_) |-----' " <<
endl;
        cout << "          `-----+'\\  / | Y  A| " <<
endl;
        cout << "          |  \\/ A|-----' " <<
endl;
        cout << "          `-----' " <<
endl;
        cout << "-----" <<
endl;
        cout << "          created by: Muhamad Salman Hakim Alfarisi " <<
endl;
        cout << "-----" <<
endl;
    }

double convertInputs(string x){
    double y;
    if (x == "A"){
        y = 1.0;
    }
    if (x == "J"){
        y = 11.0;
    }
    if (x == "Q"){
        y = 12.0;
    }
    if (x == "K"){
        y = 13.0;
    }
    if (x == "2"){
        y = 2.0;
    }
    if (x == "3"){
        y = 3.0;
    }
    if (x == "4"){
        y = 4.0;
    }
    if (x == "5"){
        y = 5.0;
    }
    if (x == "6"){
        y = 6.0;
    }
}

```

```

    }

    if (x == "7") {
        y = 7.0;
    }

    if (x == "8") {
        y = 8.0;
    }

    if (x == "9") {
        y = 9.0;
    }

    if (x == "10") {
        y = 10.0;
    }

    return y;
}

bool validInputs(string x) {
    if (x == "A" || x == "J" || x == "Q" || x == "K" || x == "2" || x ==
"3" || x == "4" || x == "5" || x == "6" || x == "7" || x == "8" || x ==
"9" || x == "10") {
        return true;
    } else {
        return false;
    }
}

double zeroDiv(int a, int b) {
    if (b == 0) {
        return 0;
    } else {
        return (double) a/b;
    }
}

double calc(double a, double b, int op) {
    if (op == 1) {
        return a + b;
    } else if (op == 2) {
        return a - b;
    } else if (op == 3) {
        return a * b;
    } else {
        return zeroDiv(a, b);
    }
}

```

```

}

bool checkDuplicate(vector<string> result, string temporary){
    for (int i = 0; i < result.size(); i++) {
        if (result[i] == temporary){
            return true;
        }
    }
    return false;
}

void printResult(vector<string> result){
    for (int i = 0; i < result.size(); i++) {
        cout << result[i] << endl;
    }
}

string exp(int sign){
    if (sign == 1){
        return "+";
    } else if (sign == 2){
        return "-";
    } else if (sign == 3){
        return "*";
    } else {
        return "/";
    }
}

void totalSolutions(vector<string> result){
    if (result.size() == 0){
        cout << "No solution" << endl;
    } else {
        printResult(result);
        cout << "Total : " << result.size() << " solutions" << endl;
    }
}

void writeToFile(vector<string> result, string input1, string input2,
string input3, string input4, double runtime){
    string fileName, filePath;
    cout << "Enter the name of the file to save the solutions: ";
    cin >> fileName;
    filePath = "../test/" + fileName;
}

```

```

    ofstream File(filePath + ".txt");
    File << "*****" << endl;
    File << "        Make It 24!        " << endl;
    File << "Solutions for " << input1 << "," << input2 << "," << input3
<< "," << input4 << endl;
    File << "Total : " << result.size() << " solutions" << endl;
    File << fixed << setprecision(3);
    File << "Runtime : " << runtime << " seconds" << endl;
    File << "*****" << endl;
    for (int i = 0; i < result.size(); i++){
        File << result[i] << endl;
    }
    File.close();
}

```

c. solver24.hpp

```

#include <iostream>
#include <time.h>
#include <vector>
#include <fstream>
#include <iomanip>
#include "operations.hpp"

using namespace std;

#ifndef SOLVER24_HPP
#define SOLVER24_HPP

/* Brute Force Applications */
void solver24(int a, int b, int c, int d, vector <string> *result);
/*I.S. Recieve 4 integer and 1 vector*/
/*F.S. Push all possible solution to vector*/

#endif

```

d. solver24.cpp

```

#include "solver24.hpp"

void solver24(int a, int b, int c, int d, vector <string> *result){
    double res; // result of calculation
    int arr[4] = {a, b, c, d}; // array of input
    string tempResult; // temporary result
    for(int i = 0; i < 4; i++){ // loop input
        for(int j = 0; j < 4; j++){
            for(int k = 0; k < 4; k++){

```



```

        for(int l = 0; l < 4; l++){
            for(int op1 = 1; op1 < 5; op1++){ // loop operator
                for(int op2 = 1; op2 < 5; op2++){
                    for(int op3 = 1; op3 < 5; op3++){
                        if (i != j && i != k && i != l && j != k && j != l && k != l){
// handle duplicate of input loop

                            // calculate all possible combination of brackets
                            // type 1
                            // (a op1 b) op2 (c op3 d)
                            res = calc(calc(arr[i], arr[j], op1), calc(arr[k], arr[l],
op3), op2);

                            if (res == 24){
                                tempResult = "(" + to_string(arr[i]) + exp(op1) +
to_string(arr[j]) + ")" + exp(op2) + "(" + to_string(arr[k]) + exp(op3) + to_string(arr[l]) +
)";

                                if (!checkDuplicate(*result, tempResult)){
                                    result->push_back(tempResult);
                                }
                            }
                            // type 2
                            // ((a op1 b) op2 c) op3 d
                            res = calc(calc(calc(arr[i], arr[j], op1), arr[k], op2),
arr[l], op3);

                            if (res == 24){
                                tempResult = "(" + to_string(arr[i]) + exp(op1) +
to_string(arr[j]) + ")" + exp(op2) + to_string(arr[k]) + ")" + exp(op3) + to_string(arr[l]);
                                if (!checkDuplicate(*result, tempResult)){
                                    result->push_back(tempResult);
                                }
                            }
                            // type 3
                            // (a op1 (b op2 c)) op3 d
                            res = calc(calc(arr[i], calc(arr[j], arr[k], op2), op1),
arr[l], op3);

                            if (res == 24){
                                tempResult = "(" + to_string(arr[i]) + exp(op1) + "(" +
to_string(arr[j]) + exp(op2) + to_string(arr[k]) + ")" + exp(op3) + to_string(arr[l]);
                                if (!checkDuplicate(*result, tempResult)){
                                    result->push_back(tempResult);
                                }
                            }
                            // type 4
                            // a op1 ((b op2 c) op3 d)
                            res = calc(arr[i], calc(calc(arr[j], arr[k], op2), arr[l],
op3), op1);

                            if (res == 24){
                                tempResult = to_string(arr[i]) + exp(op1) + "(" +
to_string(arr[j]) + exp(op2) + to_string(arr[k]) + ")" + exp(op3) + to_string(arr[l]) +
)";
                                if (!checkDuplicate(*result, tempResult)){
                                    result->push_back(tempResult);
                                }
                            }
                            // type 5
                            // a op1 (b op2 (c op3 d))
                            res = calc(arr[i], calc(arr[j], calc(arr[k], arr[l], op3),
op2), op1);

```



```

    } else if (inputF == 2){
        cout << "Thank you for using Make It 24!" << endl;
        exit(0);
    }
}
if (input == 2){
    //random inputs
    srand(time(NULL));
    a = rand() % 13 + 1; b = rand() % 13 + 1; c = rand() % 13 + 1; d = rand() % 13
+ 1;

    cout << "Your numbers are: " << a << " " << b << " " << c << " " << d << endl;

    // runtime
    clock_t start = clock();
    solver24(a, b, c, d, &result);
    totalSolutions(result);
    double runtime = double(clock() - start) / double(CLOCKS_PER_SEC);
    printf("Runtime : %.3fs\n", runtime);

    cout << "Do you want to save the solutions to a file? (1 for yes, 2 for no): ";
    cin >> inputF;
    if (inputF == 1){
        writeToFile(result, to_string(a), to_string(b), to_string(c), to_string(d),
runtime);
    } else if (inputF == 2){
        cout << "Thank you for using Make It 24!" << endl;
        exit(0);
    }
}
if (input == 3){
    cout << "Thank you for using Make It 24!" << endl;
    exit(0);
}
}
}

```

BAB III

Dokumentasi Test Case Program

a. Tampilan awal program


```

-----
                Welcome to MAKE IT 24!
This program will solve the Make It 24 game for you.

      .----- .
      |A_ _|   |A_ |
      |( \ / )|---| ( \ / )|
      | \ / | \ / | ( ) |
      | \ / A / \ | _x_ |
      +-----+ \ / | Y A |
      | \ / A |-----|
      |
-----

                created by: Muhamad Salman Hakim Alfarisi
-----

Enter 1 to input your own cards or 2 to generate random cards: 2
Your numbers are: 6 1 7 9
(1+7)*(9-6)
(7+1)*(9-6)
(9-6)*(1+7)
(9-6)*(7+1)
Total : 4 solutions
Runtime : 0.002s
Do you want to save the solutions to a file? (1 for yes, 2 for no):

```

d. Hasil save file solusi manual (10, 10, 10, 10)

```

Tucil1_13521010 > test > test1.txt
1 *****
2           Make It 24!
3 Solutions for 10,10,10,10
4 Total : 0 solutions
5 Runtime : 0.000 seconds
6 *****

```

e. Hasil save file solusi random (J, 2, 4, Q)

```

Tucil1_13521010 > test > test2.txt
1 *****
2           Make It 24!
3 Solutions for 11,2,4,12
4 Total : 2 solutions
5 Runtime : 0.000 seconds
6 *****
7 ((11-2)*4)-12
8 (4*(11-2))-12

```

f. Hasil save file solusi random (J, 9, K, 4)

```

Tucil1_13521010 > test > test4.txt
1 *****
2      Make It 24!
3      Solutions for 11,9,13,4
4      Total : 0 solutions
5      Runtime : 0.000 seconds
6      *****

```

g. Hasil save file solusi manual (6, 6, 6, 6)

```

Tucil1_13521010 > test > test6.txt
1 *****
2      Make It 24!
3      Solutions for 6,6,6,6
4      Total : 7 solutions
5      Runtime : 0.000 seconds
6      *****
7      (6+6)+(6+6)
8      ((6+6)+6)+6
9      (6+(6+6))+6
10     6+((6+6)+6)
11     6+(6+(6+6))
12     (6*6)-(6+6)
13     ((6*6)-6)-6

```

Lampiran

a. Link Repository Github

Link : https://github.com/archmans/Tucil1_13521010.git

b. Checklist

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