

# LAPORAN

Tugas Kecil 1 IF2211  
Strategi Algoritma Semester II tahun 2022/2023  
“Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force”

KELAS 01



DISUSUN OLEH

(13521065) Mutawally Nawwar

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**SEMESTER II TAHUN 2022-2023**

# DAFTAR ISI

## Contents

<b>DAFTAR ISI</b> .....	2
<b>BAB I</b> .....	3
<b>1.1 Brute Force</b> .....	3
1.1.1. Kelebihan Brute Force .....	3
1.1.2. Kekurangan Brute Force .....	3
<b>1.2. Teknik Brute Force</b> .....	3
1.2.1. Exhaustive Search .....	3
1.2.2. Heuristik Search .....	4
<b>BAB II</b> .....	5
<b>2.1. Source Code</b> .....	5
<b>BAB III</b> .....	16
<b>3.1.</b> .....	16
<b>3.2.</b> .....	16
<b>3.3.</b> .....	17
<b>3.4.</b> .....	17
<b>3.5.</b> .....	17
<b>3.6.</b> .....	18

# **BAB I**

## **Teori Dasar**

### **1.1 Brute Force**

Algoritma brute force merupakan algoritma yang menggunakan pendekatan lempang (straightforward) untuk memecahkan suatu persoalan. algoritma brute force sangat banyak penerapannya di berbagai persoalan, seperti bubble sort, selection sort, uji bilangan prima, dan lain-lain. Algoritma ini memiliki beberapa karakteristik sebagai berikut:

- a. Algoritma brute force umumnya tidak “cerdas” dan tidak mangkus, karena ia membutuhkan volume komputasi yang besar dan waktu yang lama dalam penyelesaiannya. Kadang-kadang algoritma brute force disebut juga algoritma naif (naïve algorithm).
- b. Algoritma brute force lebih cocok untuk persoalan yang ukuran masukannya (n) kecil. Algoritma brute force sering digunakan sebagai basis pembanding dengan algoritma lain yang lebih mangkus.
- c. Meskipun bukan metode problem solving yang mangkus, hampir semua persoalan dapat diselesaikan dengan algoritma brute force. Ini adalah kelebihan brute force.

#### **1.1.1. Kelebihan Brute Force**

Algoritma brute force memiliki berbagai kelebihan sebagai berikut:

- a. Algoritma brute force dapat diterapkan untuk memecahkan hampir sebagian besar masalah (wide applicability).
- b. Algoritma brute force sederhana dan mudah dimengerti.
- c. Algoritma brute force menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokan string, perkalian matriks.
- d. Algoritma brute force menghasilkan algoritma baku (standard) untuk tugas-tugas komputasi seperti penjumlahan/perkalian n buah bilangan, menentukan elemen minimum atau maksimum di dalam senarai (larik).

#### **1.1.2. Kekurangan Brute Force**

Algoritma brute force memiliki berbagai kekurangan sebagai berikut:

- a. Algoritma brute force jarang menghasilkan algoritma yang mangkus.
- b. Algoritma brute force umumnya lambat untuk masukan berukuran besar sehingga tidak dapat diterima.
- c. Tidak sekonstruktif/sekreatif strategi pemecahan masalah lainnya.

### **1.2. Teknik Brute Force**

#### **1.2.1. Exhaustive Search**

Exhaustive search adalah teknik pencarian solusi secara solusi brute force untuk persoalan-persoalan kombinatorik, yaitu persoalan di antara objek-objek kombinatorik seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan.

#### 1.2.2. Heuristik Search

Algoritma exhaustive search dapat diperbaiki kinerjanya sehingga tidak perlu melakukan pencarian solusi dengan mengeksplorasi semua kemungkinan solusi. teknik heuristik digunakan untuk mengeliminasi beberapa kemungkinan solusi tanpa harus mengeksplorasi seluruh kemungkinan solusi secara penuh.

## BAB II

### Source Program

#### 2.1. Source Code

```
#include <iostream>
#include <time.h>
#include <stdlib.h>
#include <fstream>
#include <string>
#include <chrono>
#include <algorithm>

using namespace std;

void generator(int i);
string intToChar(int x);

string arrNumber[13] = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K"};
int arrInput[4],
    arrTmp[4],
    arrResult[7680][9], // 0-3 kombinasi angka, 4-6 kombinasi operator, 7 jenis kurung [1-5],
8 apakah solusi 0/1
    cntSolution = 0,
    idx = 0;

bool isExist()
{
    bool res, same;
    res = false;
    for (int i = 0; i < idx; i++)
    {
        same = true;
        for (int j = 0; j < 4; j++)
        {
            if (arrResult[i][j] != arrTmp[j])
            {
                same = false;
                break;
            }
        }
    }

    res = same;
    if (res)
    {
```

```

        break;
    }
}

return res;
}

float calculator(float a, float b, int op)
{
    if (op == 1)
    {
        return (float)a + b;
    }
    else if (op == 2)
    {
        return (float)a - b;
    }
    else if (op == 3)
    {
        return (float)a * b;
    }
    else
    {
        return (float)a / b;
    }
}

char expr1(int x)
{
    if (x == 1)
    {
        return '+';
    }
    else if (x == 2)
    {
        return '-';
    }
    else if (x == 3)
    {
        return '*';
    }
    else
    {
        return '/';
    }
}

```

```

void checker(int i)
{
    float res, tmp;
    switch (arrResult[i][7])
    {
        case 1: //(12)(34)
            res = calculator(arrResult[i][2], arrResult[i][3], arrResult[i][6]);
            tmp = calculator(arrResult[i][0], arrResult[i][1], arrResult[i][4]);

            res = calculator(tmp, res, arrResult[i][5]);

            break;
        case 2: //((12)3)4
            res = calculator(arrResult[i][0], arrResult[i][1], arrResult[i][4]);
            res = calculator(res, arrResult[i][2], arrResult[i][5]);
            res = calculator(res, arrResult[i][3], arrResult[i][6]);

            break;
        case 3: //(1(23))4
            res = calculator(arrResult[i][1], arrResult[i][2], arrResult[i][5]);
            res = calculator(arrResult[i][0], res, arrResult[i][4]);
            res = calculator(res, arrResult[i][3], arrResult[i][6]);

            break;
        case 4: // 1((23)4)
            res = calculator(arrResult[i][1], arrResult[i][2], arrResult[i][5]);
            res = calculator(res, arrResult[i][3], arrResult[i][6]);
            res = calculator(arrResult[i][0], res, arrResult[i][4]);

            break;
        case 5: // 1(2(34))
            res = calculator(arrResult[i][2], arrResult[i][3], arrResult[i][6]);
            res = calculator(arrResult[i][1], res, arrResult[i][5]);
            res = calculator(arrResult[i][0], res, arrResult[i][4]);

            break;
    }

    if ((float)(res - 24.0) <= 0.00001 && (float)(res - 24.0) >= -0.00001)
    {
        arrResult[i][8] = 1;
        cntSolution++;
    }
}

```

```

void bruteForce24game()
{
    for (int i = 0; i < 4; i++)
    { // loop angka
        for (int j = 0; j < 4; j++)
        {
            if (j != i)
            {
                for (int k = 0; k < 4; k++)
                {
                    if (k != i && k != j)
                    {
                        for (int l = 0; l < 4; l++)
                        {
                            if (l != i && l != j && l != k)
                            {
                                arrTmp[0] = arrInput[i];
                                arrTmp[1] = arrInput[j];
                                arrTmp[2] = arrInput[k];
                                arrTmp[3] = arrInput[l];
                                if (isExist())
                                {
                                    break;
                                }
                                else
                                {
                                    for (int m = 1; m < 5; m++)
                                    { // loop operator
                                        for (int n = 1; n < 5; n++)
                                        {
                                            for (int o = 1; o < 5; o++)
                                            {
                                                for (int p = 1; p <= 5; p++)
                                                {
                                                    arrResult[idx][0] = arrInput[i];
                                                    arrResult[idx][1] = arrInput[j];
                                                    arrResult[idx][2] = arrInput[k];
                                                    arrResult[idx][3] = arrInput[l];
                                                    arrResult[idx][4] = m;
                                                    arrResult[idx][5] = n;
                                                    arrResult[idx][6] = o;
                                                    arrResult[idx][7] = p;
                                                    arrResult[idx][8] = 0;
                                                    idx++;
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```



```

    }
    }
    }
    }
    }
    }
    }
    }
    }
    }
}

void generator(int i)
{
    string res;
    if (arrResult[i][8] == 1)
    {
        switch (arrResult[i][7])
        {
            case 1: //(12)(34)
                res =
                '('+intToChar(arrResult[i][0])+expr1(arrResult[i][4])+intToChar(arrResult[i][1])+')'+expr1(arrResult[i][5])+
                '('+intToChar(arrResult[i][2])+expr1(arrResult[i][6])+intToChar(arrResult[i][3])
                )+');

                break;
            case 2: //((12)3)4
                res =
                "("+intToChar(arrResult[i][0])+expr1(arrResult[i][4])+intToChar(arrResult[i][1])+')'+expr1(arrResult[i][5])+
                intToChar(arrResult[i][2])+')'+expr1(arrResult[i][6])+intToChar(arrResult[i][3]);

                break;
            case 3: //(1(23))4
                res =
                '('+intToChar(arrResult[i][0])+expr1(arrResult[i][4])+')'+intToChar(arrResult[i][1])+expr1(arrResult[i][5])+
                intToChar(arrResult[i][2])+')'+expr1(arrResult[i][6])+intToChar(arrResult[i][3]);

                break;
            case 4: //1((23)4)
                res =
                intToChar(arrResult[i][0])+expr1(arrResult[i][4])+("("+intToChar(arrResult[i][1])+expr1(arrResult[i][5])+
                intToChar(arrResult[i][2])+')'+expr1(arrResult[i][6])+intToChar(arrResult[i][3])
                )+');

```

```

        break;
    case 5: // 1(2(34))
        res =
intToChar(arrResult[i][0])+expr1(arrResult[i][4])+'('+intToChar(arrResult[i][1])+expr1(arrResult[i][5])+'('+intToChar(arrResult[i][2])+')'+expr1(arrResult[i][6])+intToChar(arrResult[i][3]))+')';

        break;
    }
    cout << res << endl;
}
}

bool isIn(string x){
    bool inside;
    inside = false;
    for(int i=0;i<13;i++){
        if(x == arrNumber[i]){
            inside = true;
            break;
        }
    }
    return inside;
}

int charToInt(string x){
    int res;
    for(int i=0;i<13;i++){
        if(arrNumber[i] == x){
            res = i+1;
        }
    }

    return res;
}

string intToChar(int x){
    return arrNumber[x-1];
}

void inputFile(){
    string fileName, myText, tmp, tes;
    int length, cnt;
    bool isValid;

    cout << "Masukkan nama file input: ";

```

```

cin >> fileName;

isValid = false;
while(!isValid){
    ifstream MyReadFile("../test/"+fileName+".txt");
    getline(MyReadFile, myText);
    MyReadFile.close();

    length = myText.length();
    cnt = 0;
    tmp = "";
    isValid = true;
    for(int i=0;i<=length;i++){
        if(myText[i] != ' ' && i != length){
            tmp += myText[i];
        }else{
            if(!isIn(tmp)){
                isValid = false;
                break;
            }else{
                arrInput[cnt] = charToInt(tmp);
                cnt++;
                tmp = "";
            }
        }
        if(int(myText[i]) == 0){
            break;
        }
    }

    if(cnt != 4){
        isValid = false;
    }

    if(!isValid){
        cout << "Pastikan format benar dan karakter input diantara (A 2 3 4 5 6 7 8 9 10 J Q
K)" << endl;
        cout << "ketik apapun untuk melakukan pembacaan file..." << endl;
        cin >> tes;
    }
}

void inputKeyboard(){
    string tmp, inputLine;
    char tmpInput;

```

```

bool isValid;
int cnt, length;

getline(cin, inputLine);
isValid = false;
while(!isValid){
    cout << "Masukan input dengan format '1 2 3 4'" << endl;

    getline(cin, inputLine);

    length = inputLine.length();
    cnt = 0;
    tmp = "";
    isValid = true;
    for(int i=0;i<=length;i++){
        if(inputLine[i] != ' ' && i != length){
            tmp += inputLine[i];
        }else{
            if(!isIn(tmp)){
                isValid = false;
                break;
            }else{
                arrInput[cnt] = charToInt(tmp);
                cnt++;
                tmp = "";
            }
        }
        if(int(inputLine[i]) == 0){
            break;
        }
    }

    if(cnt != 4){
        isValid = false;
    }

    if(!isValid){
        cout << "Pastikan format benar dan karakter input diantara (A 2 3 4 5 6 7 8 9 10 J Q
K)" << endl << endl;
    }
}

void inputRandom(){
    int n = 13;
    srand(time(NULL));

```

```

for (int i = 0; i < 4; i++)
{
    arrInput[i] = (rand() % n) + 1;
    cout << arrInput[i] << " ";
}
cout << endl;
}

void outputFile(){
    string namaFile, res;

    cout << "Masukan nama file untuk menyimpan solusi" << endl;
    cin >> namaFile;

    ofstream MyFile("../test/" + namaFile + ".txt");

    // Write to the file
    for(int i=0;i<idx;i++){
        if(arrResult[i][8] == 1){
            switch (arrResult[i][7])
            {
                case 1: //(12)(34)
                    res =
'('+intToChar(arrResult[i][0])+expr1(arrResult[i][4])+intToChar(arrResult[i][1])+')'+expr1(arr
Result[i][5])+('+intToChar(arrResult[i][2])+expr1(arrResult[i][6])+intToChar(arrResult[i][3]
))+');

                    break;
                case 2: //((12)3)4
                    res =
"('"+intToChar(arrResult[i][0])+expr1(arrResult[i][4])+intToChar(arrResult[i][1])+')'+expr1(a
rrResult[i][5])+intToChar(arrResult[i][2])+')'+expr1(arrResult[i][6])+intToChar(arrResult[i][
3]));

                    break;
                case 3: //(1(23))4
                    res =
'('+intToChar(arrResult[i][0])+expr1(arrResult[i][4])+('+intToChar(arrResult[i][1])+expr1(arr
Result[i][5])+intToChar(arrResult[i][2])+')'+expr1(arrResult[i][6])+intToChar(arrResult[i][
3]));

                    break;
                case 4: // 1((23)4)
                    res =
intToChar(arrResult[i][0])+expr1(arrResult[i][4])+("('"+intToChar(arrResult[i][1])+expr1(arrR

```

```

result[i][5])+intToChar(arrResult[i][2])+')'+expr1(arrResult[i][6])+intToChar(arrResult[i][3]
+');

        break;
        case 5: // 1(2(34))
            res =
intToChar(arrResult[i][0])+expr1(arrResult[i][4])+(''+intToChar(arrResult[i][1])+expr1(arrRe
sult[i][5])+(''+intToChar(arrResult[i][2])+')'+expr1(arrResult[i][6])+intToChar(arrResult[i][3
]))+"))";

            break;
        }
        MyFile << res << '\n';
    }
}

// Close the file
MyFile.close();
}

int main()
{
    int choosen;

    cout << "Silakan pilih metode inputan: " << endl;
    cout << "1. Input Keyboard" << endl;
    cout << "2. Input File" << endl;
    cout << "Ketik selain 1,2 jika memilih Random" << endl;
    cin >> choosen;

    switch (choosen)
    {
        case 1:
            inputKeyboard();
            break;
        case 2:
            inputFile();
            break;
        case 3:
            inputRandom();
            break;
    }

    auto START = chrono::high_resolution_clock::now();
    bruteForce24game();
    for (int i = 0; i < idx; i++)

```

```

{
    checker(i);
}
cout << cntSolution << " solusi ditemukan" << endl;
for (int i = 0; i < idx; i++)
{
    generator(i);
}

cout << "Apakah ingin menyimpan solusi ke dalam file?" << endl;
cout << "1. ya" << endl;
cout << "ketik selain 1,jika tidak" << endl;

cin >> choosen;

if(choosen == 1){
    outputFile();
}

auto STOP = chrono::high_resolution_clock::now();
auto duration = chrono::duration_cast<chrono::microseconds>(STOP - START);
cout << "Waktu eksekusi: " << (float) (duration.count()/1000) << " ms" << endl;

system("pause");
}

```

## BAB III

### Tes

3.1.

```
Masukan input dengan format '1 2 3 4'
Q J K 2
20 solusi ditemukan
Q+((J+K)/2)
Q+((K+J)/2)
((J-Q)+K)*2
(J-(Q-K))*2
((J+K)-Q)*2
(J+(K-Q))*2
((J+K)/2)+Q
((K-Q)+J)*2
(K-(Q-J))*2
((K+J)-Q)*2
(K+(J-Q))*2
((K+J)/2)+Q
2*((J-Q)+K)
2*(J-(Q-K))
2*((J+K)-Q)
2*(J+(K-Q))
2*((K-Q)+J)
2*(K-(Q-J))
2*((K+J)-Q)
2*(K+(J-Q))
Apakah ingin menyimpan solusi ke dalam file?
1. ya
ketik selain 1,jika tidak
2
Waktu eksekusi: 37230 ms
```

3.2.

```
Masukan input dengan format '1 2 3 4'
6 6 6 6
7 solusi ditemukan
(6+6)+(6+6)
((6+6)+6)+6
(6+(6+6))+6
6+((6+6)+6)
6+(6+(6+6))
(6*6)-(6+6)
((6*6)-6)-6
Apakah ingin menyimpan solusi ke dalam file?
1. ya
ketik selain 1,jika tidak
1
Masukan nama file untuk menyimpan solusi
output
Waktu eksekusi: 9382 ms
```



3.3.

```
Silakan pilih metode inputan:
1. Input Keyboard
2. Input File
Ketik selain 1,2 jika memilih Random
2
Masukkan nama file input: input
242 solusi ditemukan
((A+2)+3)*4
(A+(2+3))*4
(A*2)*(3*4)
((A*2)*3)*4
(A*(2*3))*4
A*((2*3)*4)
A*(2*(3)*4))
(A*2)*(4*3)
((A*2)*4)*3
(A*(2*4))*3
A*((2*4)*3)
A*(2*(4)*3))
((A+3)+2)*4
(A+(3+2))*4
(A+3)*(2+4)
(A*3)*(2*4)
((A*3)*2)*4
(A*(3*2))*4
A*((3*2)*4)
A*(3*(2)*4))
(A+3)*(4+2)
(A*3)*(4*2)
((A*3)*4)*2
```

3.4.

```
Masukan input dengan format '1 2 3 4'
10 8 3 3
61 solusi ditemukan
(10+8)+(3+3)
((10+8)+3)+3
(10+(8+3))+3
10+((8+3)+3)
10+(8+(3)+3))
(10+3)+(8+3)
((10+3)+8)+3
(10+(3+8))+3
10+((3+8)+3)
10+(3+(8)+3))
(10+3)+(3+8)
((10+3)+3)+8
(10+(3+3))+8
```

3.5.

```
Masukan input dengan format '1 2 3 4'
10 5 8 3
0 solusi ditemukan
```

### 3.6.

```
Masukan input dengan format '1 2 3 4'
9 2 7 K
34 solusi ditemukan
((9*2)-7)+K
(9*2)-(7-K)
(9*2)+(K-7)
((9*2)+K)-7
((9-7)*K)-2
((2*9)-7)+K
(2*9)-(7-K)
(2*9)+(K-7)
((2*9)+K)-7
((2*K)-9)+7
(2*K)-(9-7)
(2*K)+(7-9)
((2*K)+7)-9
(7-9)+(2*K)
7-(9-(2)*K))
```

## **LAMPIRAN**

[https://github.com/mutawalle/Tucil1\\_13521065.git](https://github.com/mutawalle/Tucil1_13521065.git)