

Laporan Tugas Kecil IF2211 Strategi Algoritma

Semester II Tahun Pelajaran 2022/2023

Penyelesaian Permainan Kartu 24 dengan Algoritma *Brute Force*

Althaaf Khasyi Atisomya – 13521130 – K2

A. Algoritma *Brute Force*

Untuk mengimplementasikan algoritma brute force digunakan bahasa pemrograman C++ yang dijalankan pada *windows*. *Library* yang digunakan pada program ini adalah *iostream*, *sstream*, *fstream*, *vector*, *ctime*, *cstdlib*, dan *chrono*. *Library* *iostream* dan *sstream* digunakan untuk keperluan input pengguna dan validasinya. *Library* *fstream* digunakan untuk menyimpan hasil perhitungan dalam bentuk file. *Library* *vector* untuk keperluan struktur data *vector* yang nantinya akan digunakan sebagai struktur data kartu, dan hasil perhitungan. *Library* *ctime* dan *cstdlib* digunakan untuk men-generate random kartu. *Library* *chrono* untuk menghitung waktu eksekusi algoritma.

Program ini dapat menerima empat kartu dari user maupun empat kartu yang ter-generate secara acak. Pada program, empat kartu tersebut direpresentasikan sebagai *vector of float*. Algoritma *brute force* dilakukan dengan mengecek semua kemungkinan kejadian yang ada. Dalam permainan ini variasi kemungkinan kejadian dilakukan dengan variasi urutan kartu, variasi operator (+, -, *, dan /), dan variasi tanda kurung. Terdapat lima variasi tanda kurung, yaitu $((A+A) + A) + A$, $(A+A) + (A+A)$, $(A+ (A+A)) + A$, $A+ ((A+A) + A)$, dan $A+ (A+ (A+A))$. Untuk variasi urutan kartu, kartu yang sama tidak dapat muncul dua kali sehingga banyaknya permutasi menjadi $4 \times 3 \times 2 \times 1$. Pada variasi operator, operator yang sama dapat dipakai berkali-kali sehingga permutasinya menjadi $4 \times 4 \times 4$. Oleh karena itu, terdapat $4 \times 3 \times 2 \times 1 \times 4 \times 4 \times 4 \times 5$ kejadian yang harus ditangani.

Perhitungan operasi matematika tiap kejadian dilakukan oleh fungsi `calculate(variant, num1, num2, num3, num4, op1, op2, op3)`. Parameter `variant` pada fungsi `calculate` melambangkan varian tanda kurung antara 1-5. Parameter `num1`, `num2`, `num3`, dan `num4` adalah empat kartu yang dimiliki. Parameter `op1`, `op2`, dan `op3` adalah operator yang akan digunakan dalam operasi matematika (+, -, *, dan /). Pada fungsi `makeIt24` inilah dilakukan

pemanggilan fungsi calculate dengan semua variasi urutan angka, variasi operator, dan variasi tanda kurung, apabila fungsi calculate mengembalikan angka 24, maka operasi matematika tersebut akan dimasukkan pada vektor hasil. Apabila kartu terdapat setidaknya 2 kartu yang sama maka terdapat kemungkinan 2 operasi matematika yang sama masuk dalam hasil perhitungan, untuk mencegah hal ini dilakukan pengecekan apakah operasi matematika tersebut telah terdapat pada vector hasil atau belum sehingga tidak akan menghasilkan dua operasi matematika yang sama.

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

B. Source Program

1. File main.cpp

[illegible]

```

        cout << "\nIngin menyimpan hasil?\n" <<
        "    1. Ya\n" <<
        "    2. Tidak\n" <<
        ">> ";
        getline(cin, str, '\n');
    }
    int opt = stoi(str);
    switch (opt){
        case 1:
            outputFile(*result);
            break;
        case 2:
            break;
        default:
            break;
    }
}

void inputMenu(bool *const &start, vector<string> *const &result){
    // menampilkan menu input
    cout << "\nPilih metode input:\n" <<
    "    1. Input manual\n" <<
    "    2. Input random\n" <<
    "    3. Keluar\n" <<
    ">> ";
    string str;
    getline(cin, str);
    while (!isInputValid(3, str)){
        cout << "\nInput salah, masukkan angka 1/2/3\n";
        cout << "\nPilih metode input:\n" <<
        "    1. Input manual\n" <<
        "    2. Input random\n" <<
        "    3. Keluar\n" <<
        ">> ";
        getline(cin, str);
    }
    int opt = stoi(str);
    switch (opt){
        case 1: //input user
            cout << "\nMasukkan 4 kartu (2-10, A, J, Q, K)\n" <<
            ">> ";
            *result = makeIt24(userInput());
            displayResult(*result);
            outputMenu(start, result);
            break;
        case 2: //input random
            *result = makeIt24(randomInput());
            displayResult(*result);
            outputMenu(start, result);
            break;
        case 3: //keluar dari program
            cout << "\nProgram selesai\n";
            *start = false;
            break;
        default:
            break;
    }
}

int main(){
    bool start = true;
    srand(time(0));
    vector<string> result;
    displayHomePage();
    while (start){
        inputMenu(&start, &result);
    }
    return 0;
}

```

2. File input_output.cpp

```
#include <iostream>
#include <vector>
#include <sstream> // untuk memisah spasi pada string
#include <ctime>
#include <cstdlib> // untuk randomize
#include <fstream> // untuk save file
using namespace std;

bool isCardValid(vector<string> card){
    // Mengembalikan true apabila kartu valid dinyatakan valid
    // apabila terdapat 4 kartu dan hanya terdiri dari angka 2-10, A, J, Q, atau K
    if (card.size()!=4){
        return false;
    } else {
        string validInput[13] = {"2","3","4","5","6","7","8","9","10","J","Q","K","A"};
        int isValid;
        for (int i=0; i<card.size();i++){
            isValid = 0;
            for (int j=0; j<13; j++){
                if (card[i]==validInput[j]){
                    isValid = 1;
                }
            }
            if (!isValid){
                return false;
            }
        }
        return true;
    }
}

vector<float> strCardToNum(vector<string> strCard){
    // mengubah kartu berbentuk vektor string menjadi vector float
    // A menjadi 1, J menjadi 11, Q menjadi 12, K menjadi 13
    vector<float> card;
    string x;
    for (int i=0; i<4; i++){
        if (strCard[i]=="J") card.push_back(11);
        else if (strCard[i]=="Q") card.push_back(12);
        else if (strCard[i]=="K") card.push_back(13);
        else if (strCard[i]=="A") card.push_back(1);
        else card.push_back(stoi(strCard[i]));
    }
    return card;
}

bool isInputValid(int nOption, string input){
    // mengembalikan true apabila input merupakan angka dengan range 1-nOption
    for (int i=1; i<=nOption; i++){
        if (to_string(i) == input){
            return true;
        }
    }
    return false;
}

vector<float> userInput(){
    // mengembalikan 4 kartu inputan user yang berupa vector of float
    // kartu telah valid
    string str;
    vector<string> strCard;
    string x;
    getline(cin,str);

    istringstream iss(str);
```

```

        while(iss >> x){
            strCard.push_back(x);
        }
        //pengecekan ke-valid-an kartu
        while(!isCardValid(strCard)){
            strCard.clear();
            cout << "\nMaaf Kartu tidak valid\n>> " ;
            getline(cin,str);
            istream iss(str);
            while(iss >> x){
                strCard.push_back(x);
            }
        }
        // mengubah kartu yang berupa string menjadi float
        vector<float> card = strCardToNum(strCard);
        return card;
    }

    vector<float> randomInput(){
        // mengenerate kartu secara random
        int i,num;
        vector<float> card;
        for(int i=0; i<4; i++){
            card.push_back((rand())%13+1);
        }
        cout << "\nKartu Anda adalah ";
        string strCard[13] = {"A","2","3","4","5","6","7","8","9","10","J","Q","K"};
        for(int i=0; i<4; i++){
            cout << strCard[(int)card[i]-1] << " ";
        }
        cout << endl;
        return card;
    }

    void saveResult(string fileName, vector<string> result){
        // menyimpan hasil pada file
        fstream file;
        file.open("../test/"+fileName,ios_base::out);
        for(int i=0;i<result.size()-1;i++){
            file<<result[i]<<endl;
        }
        file.close();
    }

    void outputFile(vector<string> result){
        // menyimpan hasil dengan nama file inputan user
        string fileName;
        cout << "\nMasukkan nama file\n" <<
            ">> ";
        getline(cin,fileName);
        saveResult(fileName, result);
        cout << "\nHasil berhasil disimpan\n";
    }
}

```

3. File makeIt24.cpp

```

#include <iostream>
#include <chrono>
#include <vector>
#include <string>
using namespace std;

void displayResult(vector<string> result){
    // menampilkan hasil dari perhitungan serta execution time
    int n = result.size()-1;
}

```

```

    if (n==0){
        cout << "\nTidak ditemukan solusi\n";
    } else{
        cout << "\nDitemukan " << n << " solusi\n";
        for(int i=0; i< n; i++){
            cout << "    " << result[i] << endl;
        }
    }
    cout << "Execution time: " << result[n] << " microseconds\n";
}

float mathOp(float num1, char op, float num2){
    // mengembalikan hasil perhitungan antara operasi antara 2 bilangan
    float res;
    switch (op){
        case '+':
            res = num1+num2;
            break;
        case '-':
            res = num1-num2;
            break;
        case '*':
            res = num1*num2;
            break;
        case '/':
            if (num2==0){
                res =0;
            } else {
                res = num1/num2;
            }
            break;
        default:
            break;
    }
    return res;
}

string makeString(int variant, float num1, float num2, float num3, float num4, char op1, char op2,
char op3){
    // membentuk string untuk operasi matematika
    string str,a,b,c,d;
    a=to_string((int)num1);
    b=to_string((int)num2);
    c=to_string((int)num3);
    d=to_string((int)num4);
    switch (variant){
        case 1: // ((a+b)+c)+d
            str = "("+a+op1+b+")"+op2+c+")"+op3+d;
            break;
        case 2: // (a+b)+(c+d)
            str = "("+a+op1+b+")"+op2+"("+c+op3+d+")";
            break;
        case 3: // (a+(b+c))+d
            str = "("+a+op1+"("+b+op2+c+")"+op3+d;
            break;
        case 4: // a+((b+c)+d)
            str = a+op1+"("+b+op2+c+")"+op3+d+")";
            break;
        case 5: // a+(b+(c+d))
            str = a+op1+"("+b+op2+"("+c+op3+d+")";
            break;
        default:
            break;
    }
    return str;
}

float calculate(int variant, float num1, float num2, float num3, float num4, char op1, char op2,
char op3){

```

```

// mengembalikan hasil operasi matematika berdasarkan variant tanda kurung,
// 4 angka dan 3 operator
float result;
string strRes;
switch (variant){
    case 1: // ((a+b)+c)+d
        result = mathOp(mathOp(mathOp(num1,op1,num2),op2,num3),op3,num4);
        break;
    case 2: // (a+b)+(c+d)
        result = mathOp(mathOp(num1,op1,num2),op2,mathOp(num3,op3,num4));
        break;
    case 3: // (a+(b+c))+d
        result = mathOp(mathOp(num1,op1,mathOp(num2,op2,num3)),op3,num4);
        break;
    case 4: // a+((b+c)+d)
        result = mathOp(num1,op1,mathOp(mathOp(num2,op2,num3),op3,num4));
        break;
    case 5: // a+(b+(c+d))
        result = mathOp(num1,op1,mathOp(num2,op2,mathOp(num3,op3,num4)));
        break;
    default:
        break;
}
return result;
}

bool isMember(vector<string> vec, string str){
    // mengecek apakah suatu string telah terdapat pada vector of string
    for (int i=0; i<vec.size();i++){
        if (str == vec[i]){
            return true;
        }
    }
    return false;
}

vector<string> makeIt24(vector<float> card){
    // Algoritma brute force, mengecek semua kemungkinan kejadian yang ada,
    // apabila suatu operasi matematika menghasilkan 24 akan dimasukkan dalam vector hasil
    char op[4] = {'+', '-', '*', '/'};
    vector<string> result;
    string str;
    float is24;
    chrono::steady_clock::time_point begin = chrono::steady_clock::now();
    // variasi urutan angka
    for (int a=0;a<4;a++){
        for (int b=0;b<4;b++){
            if (a!=b){
                for (int c=0;c<4;c++){
                    if (c!=a && c!=b){
                        for (int d=0;d<4;d++){
                            if (d!=a && d!=b && d!=c){
                                // variasi operator
                                for(int op1=0;op1<4;op1++){
                                    for(int op2=0;op2<4;op2++){
                                        for(int op3=0;op3<4;op3++){
                                            // variasi tanda kurung
                                            for(int var=1;var<=5;var++){
                                                is24 =
calculate(var,card[a],card[b],card[c],card[d],op[op1],op[op2],op[op3])-24;
                                                if (is24>-0.000000000001 && is24<0.000000000001){
                                                    str =
makeString(var,card[a],card[b],card[c],card[d],op[op1],op[op2],op[op3]);
                                                    if (!isMember(result,str)){
                                                        result.push_back(str);
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```


2. Test 2

```
Pilih metode input:
  1. Input manual
  2. Input random
  3. Keluar
>> 2

Kartu Anda adalah 2 7 8 5

Ditemukan 4 solusi
  ((2*5)-7)*8
  8*((2*5)-7)
  8*((5*2)-7)
  ((5*2)-7)*8
Execution time: 269 microseconds

Ingin menyimpan hasil?
  1. Ya
  2. Tidak
>> 1

Masukkan nama file
>> test2.txt

Hasil berhasil disimpan
```

3. Test 3

```
Pilih metode input:
  1. Input manual
  2. Input random
  3. Keluar
>> 1

Masukkan 4 kartu (2-10, A, J, Q, K)
>> 2 6 7 10 9

Maaf Kartu tidak valid
>> 2 6 7 10

Ditemukan 8 solusi
  (2+6)*(10-7)
  ((2*7)-10)*6
  6*((2*7)-10)
  (6+2)*(10-7)
  6*((7*2)-10)
  ((7*2)-10)*6
  (10-7)*(2+6)
  (10-7)*(6+2)
Execution time: 295 microseconds

Ingin menyimpan hasil?
  1. Ya
  2. Tidak
>> 1

Masukkan nama file
>> test3.txt

Hasil berhasil disimpan
```

4. Test 4

```
Pilih metode input:
  1. Input manual
  2. Input random
  3. Keluar
>> 2

Kartu Anda adalah 10 4 4 9

Tidak ditemukan solusi
Execution time: 263 microseconds

Ingin menyimpan hasil?
  1. Ya
  2. Tidak
>> 1

Masukkan nama file
>> test4.txt
```

5. Test 5

```
Pilih metode input:
  1. Input manual
  2. Input random
  3. Keluar
>> 1

Masukkan 4 kartu (2-10, A, J, Q, K)
>> 4 J 7 Q

Ditemukan 16 solusi
  4*((11+7)-12)
  4*(11+(7-12))
  4*((11-12)+7)
  4*(11-(12-7))
  4*((7+11)-12)
  4*(7+(11-12))
  4*((7-12)+11)
  4*(7-(12-11))
  ((11+7)-12)*4
  (11+(7-12))*4
  ((11-12)+7)*4
  (11-(12-7))*4
  ((7+11)-12)*4
  (7+(11-12))*4
  ((7-12)+11)*4
  (7-(12-11))*4
Execution time: 299 microseconds

Ingin menyimpan hasil?
  1. Ya
  2. Tidak
>> 1

Masukkan nama file
>> test5.txt

Hasil berhasil disimpan
```

6. Test 6

```
Pilih metode input:
  1. Input manual
  2. Input random
  3. Keluar
>> 2

Kartu Anda adalah 9 3 7 Q

Ditemukan 30 solusi
  9-(3*(7-12))
  9+(3*(12-7))
  (9*(7-3))-12
  9-((7-12)*3)
  9+((12-7)*3)
  3-((9-12)*7)
  3-(7*(9-12))
  ((3*7)-9)+12
  (3*7)-(9-12)
  3+(7*(12-9))
  ((3*7)+12)-9
  (3*7)+(12-9)
  3+((12-9)*7)
  (3*(12-7))+9
  ((7-3)*9)-12
  ((7*3)-9)+12
  (7*3)-(9-12)
  ((7*3)+12)-9
  (7*3)+(12-9)
  (7*(12-9))+3
  (12-9)+(3*7)
  12-(9-(3*7))
  (12-9)+(7*3)
  12-(9-(7*3))
  ((12-9)*7)+3
  (12+(3*7))-9
  12+((3*7)-9)
  (12+(7*3))-9
  12+((7*3)-9)
  ((12-7)*3)+9

Execution time: 320 microseconds
```

D. Link Repository

https://github.com/althaafka/Tucil1_13521130.git