

Tugas Kecil 1 IF2211 Strategi Algoritma
Semester 2 Tahun 2022/2023

Penyelesaian Permainan Karu 24 dengan Algoritma Brute Force

Oleh :

Varraz Hazzandra Abrar

13521020

K03



Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2023

A. BRUTE FORCE

Brute Force adalah salah satu strategi algoritma dengan pendekatan lempang (*straightforward*) untuk menyelesaikan suatu masalah. Strategi ini didasarkan pada pernyataan pada persoalan dan konsep yang dilibatkan. Brute Force memecahkan masalah dengan sangat sederhana, langsung, dan jelas caranya. Beberapa contoh penerapan Brute Force adalah pencarian elemen terkecil dalam suatu senarai, pencarian rute terpendek, pengalihan dua buah matriks, dan penyelesaian permainan kartu 24.

B. Kartu 24

Permainan kartu 24 merupakan permainan mencari angka 24 dengan memasukkan empat buah kartu remi secara acak. Pemain hanya boleh menggunakan operasi kali, bagi, tambah, kurang dan memanfaatkan tanda kurung untuk diterapkan pada keempat kartu untuk mencari nilai 24.

C. Algoritma Brute Force

1. Operan dibagikan ke 3 slot yang merupakan celah di antara tiap masukan (angka)
2. Lakukan perulangan *for* pada tiap operan. Di dalam perulangan *for* terakhir, lakukan operasi kalkulasi dan penampungan solusi untuk hasil kalkulasi yang sama dengan 24. Penampungan dilakukan sekaligus menghitung total solusi dengan inkremen tiap *goal* (24) ditemukan.
3. Ada 5 kemungkinan kalkulasi berdasarkan kondisi tanda kurung : ((ab)c)d, (a(bc))d, a(b(cd)), a((bc)d), dan (ab)(cd).

D. Source Code

Main.cpp

```
#include "jikaInputValid.cpp"
using namespace std;

int n;
string c1,c2,c3,c4, op1, op2, op3;

void menu24(){
    cout << "Menu :"<<endl;
    cout << "1. Input 4 kartu"<<endl;
    cout << "2. Random dari game"<<endl;
    cout << "3. Keluar"<<endl;
}

int main(){
    while(true){
```

```

    cout << "24 Game"<<endl;
    menu24();
    cout << "Masukkan pilihan :"<<" " ;
    cin>> n;

    while(!inputMenuValidator(n)){
        cout << "Input tidak valid. Masukkan lagi sesuai menu!"<<endl;
        cin>> n;
    }

    if(n==1 or n==2){
        if(n==1){
            cout << "Masukkan 4 kartu ! " <<endl;
            cin >> c1 >> c2 >> c3 >> c4;
            if(!validInp(c1) || !validInp(c2) || !validInp(c3) ||
!validInp(c4)){
                cout << "Input tidak valid, masukkan lagi!"<< endl ;
                cin >> c1 >> c2 >> c3 >> c4;
            }
            else{
                kartu[0] = strInt(c1);kartu[1] = strInt(c2);kartu[2] =
strInt(c3);kartu[3] = strInt(c4);
                jikaInputValid();
            }
        }
        else {
            for(int i=0; i<4; i++){
                kartu[i] = rand() % 13 + 1;
            }
            cout << "Kartu yang terpilih :";
            for(int i=0; i<4; i++){
                cout<< intkartu(kartu[i]);
            }
            cout << endl;
            jikaInputValid();
        }
    }
    else if(n == 3){
        cout << "Sip!";
    }
    return 0;
}
}

```

jikaInputValid.cpp

```

#include<bits/stdc++.h>
#include <chrono>
#include "operasi24.cpp"
using namespace std;

void jikaInputValid(){
    sort(kartu, kartu+4);
    auto mulai = chrono::steady_clock::now();
    do {
        posibilitiTo24();
    }while (next_permutation(kartu, kartu+4));
    auto selesai = chrono::steady_clock::now();
    auto durasi = chrono::duration_cast<chrono::nanoseconds>(selesai -
mulai);

    if (totalSolusi == 0){
        cout<< "Solusi nihil" ;
    }
    else{
        cout << "Banyak Solusi = "<<" "<< totalSolusi << endl;
        cout << "Solusi : " <<endl ;
        for (i = 1 ; i<= totalSolusi; i++){
            cout << hasilOpr[i]<< endl ;
        }
        cout << "Waktu penyelesaian = " << durasi.count() << " " <<
"nanosekon"<<endl ;
    }

    cout << "Apakah anda ingin menyimpan hasil pencarian ke dalam
file?"<<endl;
    cout << "masukkan y atau n!"<<endl ;
    char yn;
    cin >> yn;
    while(yn != 'y' && yn != 'n'){
        cout << "Input tidak valid, masukkan lagi input!"<<endl;
        cin >> yn;
    }

    if(yn == 'y'){
        ofstream file;
        string namafile;
        cout << "Masukkan nama file: ";
        cin >> namafile;
        namafile = "../test/" + namafile + ".txt";
        file.open(namafile);
        file << "Kartu yang terpilih :";
    }
}

```

```

        for(int i=0; i<4; i++){
            file << intkartu(kartu[i]);
        }
        file << endl;
        if(totalSolusi == 0){
            file << "Solusi nihil"<< endl;
        }
        else{
            file << "Banyak solusi: " << totalSolusi << endl;
            file << "Solusi : "<< endl;
            for(int i = 1; i <= totalSolusi; i++){
                file << hasilOpr[i] << endl;
            }
        }
        file << "Waktu penyelesaian = " << durasi.count() << " " <<
endl ;
        file.close();
    }
    else{
        cout << "Hasil pencarian tidak disimpan"<<endl;
    }

    cout << "Apakah anda ingin melanjutkan permainan?"<<endl;
    cout << "y atau n?"<<endl;
    char yn2;
    cin >> yn2;
    while(yn2 != 'y' && yn2 != 'n'){
        cout << "Input tidak valid. Input lagi!"<<endl;
        cin >> yn2;
    }
    if(yn2 == 'n'){
        cout << "Sip. Terima kasih sudah bermain, ya!";
    }
}

```

Operasi24.cpp

```

#include "olahInput.cpp"
#include "mathTo24.cpp"
using namespace std;

int totalSolusi = 0;
string hasilOpr[99999] = {} ;
int i,j,k;
int kartu[4];

void posibilitiTo24(){

```

```

for (i=0; i=3; i++){
    for (j=0; j=3; j++){
        for (k=0; k=3; k++){
            string op1, op2, op3 ;
            tandaToInt(op1); tandaToInt(op2); tandaToInt(op3);
            // ((ab)c)d
            if (kalkulasi1( kartu, i, j, k)== 24){
                totalSolusi ++ ;
                hasilOpr[totalSolusi] = "(" +
intkartu(kartu[0]) + " " + op1 + " " + intkartu(kartu[1]) + ") " + op2 + " " +
intkartu(kartu[2]) + ") " + op3 + " " + intkartu(kartu[3]);
            }
            // (ab)(cd)
            if (kalkulasi2( kartu, i, j, k)== 24){
                totalSolusi ++ ;
                hasilOpr[totalSolusi] = "(" +
intkartu(kartu[0]) + " " + op1 + " " + intkartu(kartu[1]) + ") " + op2 + " ("
+ intkartu(kartu[2]) + " " + op3 + " " + intkartu(kartu[3]) + ")";
            }
            // (a(bc))d
            if (kalkulasi3( kartu, i, j, k)== 24){
                totalSolusi ++ ;
                hasilOpr[totalSolusi] = "(" +
intkartu(kartu[0]) + " " + op1 + " (" + intkartu(kartu[1]) + " " + op2 + " " +
intkartu(kartu[2]) + ")) " + op3 + " " + intkartu(kartu[3]) ;
            }
            // a((bc)d)
            if (kalkulasi4( kartu, i, j, k)== 24){
                totalSolusi ++ ;
                hasilOpr[totalSolusi] = intkartu(kartu[0]) + "
" + op1 + " (" + intkartu(kartu[1]) + " " + op2 + " " + intkartu(kartu[2]) +
") " + op3 + " " + intkartu(kartu[3]) + ")" ;
            }
            // a(b(cd))
            if (kalkulasi5( kartu, i, j, k)== 24){
                totalSolusi ++ ;
                hasilOpr[totalSolusi] = intkartu(kartu[0]) + "
" + op1 + " " + "(" + intkartu(kartu[1]) + " " + op2 + " " + "(" +
intkartu(kartu[2]) + " " + op2 + " " + intkartu(kartu[3]) + ")")" ;
            }
        }
    }
}

```

mathTo24.cpp

```
#include<bits/stdc++.h>

double perhitungan (int a, int b, int tanda){
    int hasil;
    switch (tanda){
        case 0 :
            hasil = a+b ;
        case 1 :
            hasil = a-b ;
        case 2 :
            hasil = a*b ;
        case 3 :
            if (b!=0){
                hasil = a/b ;
            }
            else{ //b=0
                hasil = 999999999;
            }
    }
    return hasil;
}

int tandaToInt(string tanda){
    if (tanda == "+"){
        return 0 ;
    }
    else if (tanda == "-"){
        return 1 ;
    }
    else if (tanda == "*"){
        return 2 ;
    }
    else if (tanda == "/"){
        return 3 ;
    }
}

string intToTanda(int operan){
    if (operan==0){
        return "+" ;
    }
    else if(operan==1){
        return "-" ;
    }
    else if(operan==2){
        return "*" ;
    }
}
```

```

        else if(operan==3){
            return "/" ;
        }
    }

// ((ab)c)d
double kalkulasi1 (int input[4], int op1, int op2, int op3){
    double hasil ;
    hasil = perhitungan(input[0], input[1], op1) ;
    hasil = perhitungan(hasil, input[2], op2) ;
    hasil = perhitungan(hasil, input[3], op3) ;
    return hasil;
}

// (ab)(cd)
double kalkulasi2 (int input[4], int op1, int op2, int op3){
    double hasil1, hasil2, hasil ;
    hasil1 = perhitungan(input[0], input[1], op1) ;
    hasil2 = perhitungan(input[2], input[3], op3) ;
    hasil = perhitungan(hasil1, hasil2, op2) ;
    return hasil;
}

// (a(bc))d
double kalkulasi3 (int input[4], int op1, int op2, int op3){
    double hasil ;
    hasil = perhitungan(input[1], input[2], op2) ;
    hasil = perhitungan(input[0], hasil, op1) ;
    hasil = perhitungan(hasil, input[3], op3) ;
    return hasil;
}

// a((bc)d)
double kalkulasi4 (int input[4], int op1, int op2, int op3){
    double hasil ;
    hasil = perhitungan(input[1], input[2], op2) ;
    hasil = perhitungan(hasil, input[3], op3) ;
    hasil = perhitungan(input[0], hasil, op1) ;
    return hasil;
}

// a(b(cd))
double kalkulasi5 (int input[4], int op1, int op2, int op3){
    double hasil ;
    hasil = perhitungan(input[2], input[3], op3) ;
    hasil = perhitungan(input[1], hasil, op2) ;
    hasil = perhitungan(input[0], hasil, op1) ;
    return hasil;}

```


olahInput.cpp

```
#include<bits/stdc++.h>
using namespace std;

bool inputMenuValidator(int n){
    return (n==1||n==2||n==3);
}

bool validInp (string typing){

    if (typing.length() == 1 ){
        return (typing[0] == 'A' ||typing[0] == '2' ||typing[0] ==
'3' ||typing[0] == '4' ||typing[0] == '5' ||typing[0] == '6' ||typing[0] ==
'7' ||typing[0] == '8' ||typing[0] == '9' ||typing[0] == 'J' ||typing[0] ==
'Q' ||typing[0] == 'K');
    }
    else if (typing.length()== 2){
        return (typing[0]=='1' && typing[1]== '0');
    }
    else{
        return false;
    }
}

int strInt (string str){
    int nilai ;
    if (str == "A"){
        nilai = 1;
    } else if (str == "2") {
        nilai = 2;
    } else if (str == "3") {
        nilai = 3;
    } else if (str == "4") {
        nilai = 4;
    } else if (str == "5"){
        nilai = 5;
    } else if (str == "6") {
        nilai = 6;
    } else if (str == "7") {
        nilai = 7;
    } else if (str == "8") {
        nilai = 8;
    }
}
```

```

    } else if (str == "9"){
        nilai = 9;
    } else if (str == "10") {
        nilai = 10;
    } else if (str == "J") {
        nilai = 11;
    } else if (str == "Q") {
        nilai = 12;
    } else if (str == "K"){
        nilai = 13;
    }
    return nilai;
}

```

```

string intkartu(double nilai){
    string remi;
    if (nilai == 1){
        remi == "A" ;
    }
    else if (nilai == 2){
        remi == "2" ;
    }
    else if (nilai == 3){
        remi == "3" ;
    }
    else if (nilai == 4){
        remi == "4" ;
    }
    else if (nilai == 5){
        remi == "5" ;
    }
    else if (nilai == 6){
        remi == "6" ;
    }
    else if (nilai == 7){
        remi == "7" ;
    }
    else if (nilai == 8){
        remi == "8" ;
    }
    else if (nilai == 9){
        remi == "9" ;
    }
    else if (nilai == 10){
        remi == "10" ;
    }
    else if (nilai == 11){
        remi == "J" ;
    }
}

```

```

    }
    else if (nilai == 12){
        remi == "Q" ;
    }
    else if (nilai == 13){
        remi == "K" ;
    }
    return remi ;
}

```

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan	✓	
2. Program berhasil <i>running</i>	✓	
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		✓

Link Repositori

https://github.com/varrazha/Tucil1_13521020

