

Tugas Kecil 1 IF2211 Strategi Algoritma

Penyelesaian permainan kartu 24 dengan algoritma Brute Force



Disusun oleh :
13521072 - Irsyad Nurwidiyanto Basuki

INSTITUT TEKNOLOGI BANDUNG
2022

DAFTAR ISI

BAB 1 DESKRIPSI MASALAH DAN ALGORITMA	1
1.1 Algoritma Brute Force	1
1.2 Permainan kartu 24	1
1.3 Algoritma penyelesaian permainan kartu 24 dengan pendekatan brute force	1
BAB 2 SOURCE CODE PROGRAM	3
3.1 Repository Program	3
3.2 Source Code Program	3
BAB 3 MASUKAN DAN LUARAN PROGRAM	13
LAMPIRAN	18

BAB 1

DESKRIPSI MASALAH DAN ALGORITMA

1.1 Algoritma Brute Force

Algoritma brute force merupakan pendekatan yang lempang (*straightforward*) untuk memecahkan suatu persoalan. Pada umumnya, algoritma *brute force* didasarkan pada pernyataan pada persoalan (*problem statement*) dan definisi atau konsep yang dilibatkan.

Algoritma brute force umumnya tidak “cerdas” dan tidak mangkus, karena ia membutuhkan jumlah komputasi yang besar dan waktu yang lama dalam penyelesaiannya. Terkadang algoritma *brute force* disebut juga algoritma naif. Meski demikian, hampir semua persoalan dapat diselesaikan dengan algoritma *brute force*. Hal ini dikarenakan *brute force* memiliki konsep penyelesaian yang sangat sederhana, langsung, dan jelas caranya.

1.2 Permainan kartu 24

Permainan kartu 24 adalah permainan kartu aritmatika dengan tujuan mencari cara untuk mengubah 4 buah angka random sehingga mendapatkan hasil akhir sejumlah 24. Permainan ini menarik cukup banyak peminat dikarenakan dapat meningkatkan kemampuan berhitung serta mengasah otak agar dapat berpikir dengan cepat dan akurat.

Permainan Kartu 24 biasa dimainkan dengan menggunakan kartu remi. Kartu remi terdiri dari 52 kartu yang terbagi menjadi empat suit (sekop, hati, keriting, dan wajik) yang masing-masing terdiri dari 13 kartu (As, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, dan King). Yang perlu diperhatikan hanyalah nilai kartu yang didapat (As, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, dan King). As bernilai 1, Jack bernilai 11, Queen bernilai 12, King bernilai 13, sedangkan kartu bilangan memiliki nilai dari bilangan itu sendiri.

Pada awal permainan moderator atau salah satu pemain mengambil 4 kartu dari dek yang sudah dikocok secara random. Permainan berakhir ketika pemain berhasil menemukan solusi untuk membuat kumpulan nilainya menjadi 24. Pengubahan nilai tersebut dapat dilakukan menggunakan operasi dasar matematika penjumlahan (+), pengurangan (-), perkalian (\times), divisi (/) dan tanda kurung (). Tiap kartu harus digunakan tepat sekali dan urutan penggunaannya bebas.

1.3 Algoritma penyelesaian permainan kartu 24 dengan pendekatan brute force

Penulis menggunakan pendekatan brute force untuk menyelesaikan permasalahan permainan kartu 24. Adapun langkah-langkahnya sebagai berikut :

1. Program akan menerima input dari user yang berupa empat kartu remi yang ingin dihitung. Empat kartu remi ini terdiri dari A, 2, 3, ..., J, Q, K, yang merupakan kartu - kartu remi.

Program akan melakukan validasi apakah terdapat empat kartu (tidak kurang dan tidak lebih) dan apakah termasuk dalam kartu remi.

2. Program akan menampung input ke struktur data yang berupa vektor agar dapat diakses oleh program
3. Untuk setiap 4 angka yang berasal dari input, akan dilakukan permutasi dengan cara rekursif dan dilakukan pengecekan agar tidak terdapat permutasi yang sama.
4. Dari tiap permutasi, program akan melakukan kombinasi untuk tiga operator aritmatika dari tiap operator (+,-,*,/) dan juga lima kombinasi kurung yang dapat dilakukan.

Berikut merupakan kombinasi yang dapat terbentuk dengan A,B,C dan D merupakan angka dan X merupakan operator

- a. $(A \times B) \times (C \times D)$
 - b. $((A \times B) \times C) \times D$
 - c. $(A \times (B \times C)) \times D$
 - d. $A \times (B \times (C \times D))$
 - e. $A \times ((B \times C) \times D)$
5. Setiap kombinasi dan permutasi yang menghasilkan angka 24 akan dianggap sebagai solusi dan disimpan di struktur data yang berupa vektor
 6. Di akhir program akan menampilkan solusi - solusi yang ada, berapa banyak solusi, dan waktu eksekusi program.

BAB 2

SOURCE CODE PROGRAM

3.1 Repository Program

Link repository : github.com/irsyadnb/Tucill_13521072

3.2 Source Code Program

```
void permutasi(vector<float> kartu, vector<vector<float> >& perm, int ctr1, int ctr2)
{
    if (ctr1 == ctr2) {
        bool found = false;
        int i = 0;

        while(!found && i < perm.size()){ //pengecekan duplikasi
            if(kartu == perm.at(i)){
                found = true;
            }
            i++;
        }

        if(!found){
            perm.push_back(kartu);
            return;
        }
    }

    for (int i = ctr1; i <= ctr2; i++) {

        int temp = kartu[ctr1];
        kartu[ctr1] = kartu[i];
        kartu[i] = temp;

        permutasi(kartu, perm, ctr1 + 1, ctr2);

        temp = kartu[ctr1];
        kartu[ctr1] = kartu[i];
        kartu[i] = temp;
    }
}
```

```

float operasi(float left, char op, float right){
    if(op == '+'){
        return left + right;
    }
    else if(op == '-'){
        return left - right;
    }
    else if(op == '*'){
        return left * right;
    }
    else if(op == '/'){
        try {
            float result = (float)left/right;
            //cout << result;
            return result;
        }
        catch (...) {
            cout << "";
        }
    }
}

```

```

int perhitungan(float a, float b, float c, float d, int savefile){
    int count = 0;
    for (char p : op){
        for (char q : op){
            for (char r : op){
                if (operasi(operasi(a, p, b), q, operasi(c, r, d)) == 24.00) {
                    if(savefile == 1){
                        out_file << "(" << a << p << b << ")" << q << "(" << c << r << d << ")" << endl;
                    }
                    cout << "(" << a << p << b << ")" << q << "(" << c << r << d << " ";
                    cout << "\n";
                    count++;
                }
                if (operasi(operasi(operasi(a, p, b), q, c), r, d) == 24.00) {
                    if(savefile == 1){
                        out_file << "(" << a << p << b << ")" << q << c << ")" << r << d << endl;
                    }
                    cout << "(" << a << p << b << ")" << q << c << ")" << r << d;
                    cout << "\n";
                    count++;
                }
                if (operasi(operasi(a, p, operasi(b, q, c)), r, d) == 24.00) {
                    if(savefile == 1){
                        out_file << "(" << a << p << "(" << b << q << c << ")" << r << d << endl;
                    }
                    cout << "(" << a << p << "(" << b << q << c << ")" << r << d;
                    cout << "\n";
                    count++;
                }
            }
        }
    }
}

```

```

        if (operasi(a, p, operasi(b, q, operasi(c, r, d))) == 24.00) {
            if(savefile == 1){
                out_file << a << p << "(" << b << q << "(" << c << r << d << ")" << endl;
            }
            cout << a << p << "(" << b << q << "(" << c << r << d << ")" << endl;
            cout << "\n";
            count++;
        }
        if (operasi(a, p, operasi(operasi(b, q, c), r, d)) == 24.00) {
            if(savefile == 1){
                out_file << a << p << "(" << b << q << c << ")" << r << d << ")" << endl;
            }
            cout << a << p << "(" << b << q << c << ")" << r << d << ")" << endl;
            cout << "\n";
            count++;
        }
    }
}
return count;
}

```

```

    ✓ bool validate(vector<float> kartu){
        float n = 0;
    ✓   for(int i =0;i<kartu.size();i++){
            n += kartu[i];
        }
        return(n<=52);
    }

```

```
bool validate(vector<float> kartu){ ...

float randomgen(){
    float awal = 0;
    float akhir = 0;

    awal = clock();
    for(int i = 0; i <4;i++){
        cout << " ";
    }
    akhir = clock();
    float angka = abs(round((akhir - awal)));
    return angka;
}
```



```

int main(){
    vector<float> kartu;
    vector<float> kartutemp;
    vector<float> kartuvalid;

    bool cek = false;
    int input1;
    int ctr = 0;
    float waktuawal, waktuakhir, durasi;

    float a, b, c, d, w, x, y ,z;

    int p = 0;
    cout<< " << endl;
    cout<< " / << endl;
    cout<< " | << endl;
    cout<< " | _ | /\\ ) _ ) /' ' ' ' , << endl;
    cout<< " | (.) | < > \\\\) // ' ' ` , << endl;
    cout<< " | ` | \\\\) \\\\) ~ | ~: + << endl;
    cout<< " | | // * \\\\) ' : << endl;
    cout<< " | | // * * \\\\) ~ . ' << endl;
    cout<< " | | . ' . _ // ` ` ` ` , , . . \\\\) << endl;
    cout<< " | | ` \\\\) _ / * + _ \\\\) # \\\\) ~ \\\\) 000 / \\\\) | << endl;
    cout<< " | | | / : \\\\) + * \\\\) _ \\\\) # \\\\) ~ \\\\) so / ! ! \\\\) << endl;
    cout<< " | | _ \\\\) : \\\\) * + \\\\) - \\\\) # \\\\) o / ! ! ! \\\\) << endl;
    cout<< " | | / < _ = : \\\\) + * / _ _ _ @ ! ! ! _ | << endl;
    cout<< " | | < _ / _ _ \\\\) * / ( / \\\\) _ _ _ _ | << endl;
    cout<< " | | | _ _ _ \\\\) / _ ) / * \\\\) . _ / > << endl;
    cout<< " | | | ! ! ! @ / * + \\\\) : = _ > _ / << endl;
    cout<< " | | | \\\\) ! ! ! / o \\\\) # \\\\) _ \\\\) + * \\\\) : : _ \\\\) << endl;
    cout<< " | | | \\\\) ! ! ! / o s \\\\) ~ \\\\) # \\\\) _ \\\\) * + \\\\) : / | << endl;
    cout<< " | | | \\\\) _ / 000 \\\\) ~ \\\\) # \\\\) _ \\\\) + * / - \\\\) << endl;
    cout<< " | | | \\\\) ' ' ` ` , , , , // . ' . << endl;
    cout<< " | | | , . - * * * // ' << endl;
    cout<< " | | | : ~ \\\\) * // << endl;
    cout<< " | | + : | \\\\) / \\\\) << endl;
    cout<< " | | , ~ ~ // _ ( \\\\) / \\\\) ; | << endl;
    cout<< " | | , ' ` ` / _ ( _ ( < > | ( _ ) | << endl;
    cout<< " | | \\\\) \\\\) << endl;
    cout<< " | << endl;
    cout<< " \\\\) / << endl;

```

```

while(!cek){
    cout << endl;
    cout << "          SELAMAT DATANG DI 24 SOLVER" << endl;
    cout << "\\Takjublah kalian dengan kecerdasan sang ratu\\" << endl;
    cout << endl;
    cout << "Ratu : \\Pilihan cara memilih kartu yang akan diberikan kepadaku\\" << endl;
    cout << "1. Random" << endl;
    cout << "2. Input kartu" << endl;
    cout << "Masukan : ";
    cin >> input1;
    if(input1 == 1 || input1 == 2){
        cek = true;
    }
    else{
        cout << "Ratu : \\Masukan salah. Ulangi\\" << endl;
    }
}

cout << endl;

if(input1 == 1){
    vector<float> kartu;
    bool isA, isJ, isQ, isK;
    bool done = false;
    float random;
    cout << "Mendapatkan angka... " << endl;
    while(!done){
        random = randomgen() - 5;
        if(random > 1 && random < 14){
            kartu.push_back(random);
        }
        if(kartu.size() == 4){
            done = true;
        }
    }
    cout << "Kartu yang didapat : (A = 1, J = 11, Q = 12, K = 13)" << endl;
    cout << kartu[0] << " " << kartu[1] << " " << kartu[2] << " " << kartu[3] << endl;
    kartuvalid = kartu;
}

```

```

else{

    bool valid = false;
    bool cukup = false;
    bool startinput = false;
    while(!cukup || !valid){
        string n = "";
        char m;
        vector<float> kartu;
        int ctr1;
        int ctr0;

        //cout << "Masukkan kartu : ";
        getline(cin, n);
        for(int i = 0; i<n.length(); i++){
            m = n[i];
            if (n[i] != ' '){
                if(m=='A'){
                    x = 1;
                }
                else if(m=='J'){
                    x = 11;
                }
                else if(m=='Q'){
                    x = 12;
                }
                else if(m=='K'){
                    x = 13;
                }
                else if(m=='2'){
                    x = 2;
                }
                else if(m=='3'){
                    x = 3;
                }
                else if(m=='4'){
                    x = 4;
                }
                else if(m=='5'){
                    x = 5;
                }
                else if(m=='6'){
                    x = 6;
                }
                else if(m=='7'){

```

```

        else if(m=='8'){
            x = 8;
        }
        else if(m=='9'){
            x = 9;
        }
        else if(m=='1'){
            if (n[i+1] == '0'){
                x = 10;
                i++;
            }
            else{
                x = 666;
            }
        }
        else{
            x = 666;
        }
        kartu.push_back(x);
    }
}

if(kartu.size() == 4 && startinput){
    cukup = true;
    valid = validate(kartu);
}

else if (startinput){
    cout << "Ratu : \"Pastikan hanya 4 kartu!\"<< endl;
    cukup = false;
    valid = validate(kartu);
}

if(!valid && startinput){
    cout << "Ratu : \"Terdapat kartu yang salah!\"<< endl;
}

if(!startinput){
    cout << "Masukan kartu : " << endl;
}

else if (!cukup || !valid){
    cout << "Masukan kartu : " << endl;
}

kartutemp = kartu;

startinput = true;
}

kartuvalid = kartutemp;
}

```

```

vector<vector<float> > perm;
int savefile;
string namafile;
int ctr2 = kartuvalid.size() - 1;
permutasi(kartuvalid, perm, 0, ctr2);

cout << endl;

cout << "Ratu : \"Apakah ingin menyimpan solusi yang ku dapat nanti ke dalam file?\"< << endl;
cout << "1. Ya" << endl;
cout << "2. Tidak" << endl;

cout << "> ";
cin >> savefile;

if(savefile==1){
    cout << "Masukan nama file : " << endl;
    cin >> namafile;
}
else{
    remove("../test/Hasil.txt");
}

```

```

waktuawal = clock();
for (auto x : perm) {
    vector<float> cards;
    for (auto y : x) {
        cards.push_back(y);
    }
    a = cards[0];
    b = cards[1];
    c = cards[2];
    d = cards[3];
    ctr += perhitungan(a,b,c,d,savefile);
}
waktuakhir = clock();
durasi = (waktuakhir - waktuawal) / CLOCKS_PER_SEC;
if(ctr != 0){
    cout << "Ratu : \"Aku berhasil mendapat \" << ctr << " solusi.\\\"\\n";
    if(savefile == 1){
        cout << "Ratu : \"Solusi berhasil disimpan di file \" << namafile << ".txt\\\"\" << endl;
        ifstream inFile("../test/Hasil.txt");
        ofstream outFile("../test/"+namafile);
        outFile << inFile.rdbuf();
        remove("../test/Hasil.txt");
    }
}
else{
    cout << "Ratu : \"Aku tidak dapat menemukan solusi.\\\"\\n";
    if(savefile==1){
        cout << "Ratu : \"Tidak melakukan penyimpanan file.\\\"\\n";
        remove("../test/Hasil.txt");
    }
}

out_file.close();

cout << fixed << "Ratu : \"Waktu eksekusi : \" << (float)durasi << " detik\\\"\" << endl;
return 0;
}

```

BAB 3

MASUKAN DAN LUARAN PROGRAM

testinput1.txt

[illegible]

```

SELAMAT DATANG DI 24 SOLVER
"Takjublah kalian dengan kecerdasan sang ratu"

Ratu : "Pilihan cara memilih kartu yang akan diberikan kepadaku!"
1. Random
2. Input kartu
> 2

Masukan kartu :
2 3 4 5

Ratu : "Apakah ingin menyimpan solusi yang ku dapat nanti ke dalam file?"
1. Ya
2. Tidak
> 1

Masukan nama file :
testinput1

```

```

2*(3+(4+5))
2*((3+4)+5)
2*(3+(5+4))
2*((3+5)+4)
2*(4+(3+5))
2*((4+3)+5)
2*(4+(5+3))
2*((4+5)+3)
2*(5+(4+3))
2*((5+4)+3)
2*(5+(3+4))
2*((5+3)+4)
((3-2)+5)*4
(3-(2-5))*4
((3+4)+5)*2
(3+(4+5))*2
((3+5)+4)*2
(3+(5+4))*2
((3+5)-2)*4
(3+(5-2))*4
4*((3-2)+5)
4*(3-(2-5))
((4+3)+5)*2
(4+(3+5))*2
4*(3+(5-2))
4*((3+5)-2)
4*((5-2)+3)
4*(5-(2-3))
((4+5)+3)*2
(4+(5+3))*2
4*(5+(3-2))
4*((5+3)-2)
((5+3)+4)*2
(5+(3+4))*2
((5+3)-2)*4
(5+(3-2))*4
((5+4)+3)*2
(5+(4+3))*2
((5-2)+3)*4
(5-(2-3))*4
Ratu : "Aku berhasil mendapat 40 solusi."
Ratu : "Solusi berhasil disimpan di file testinput1.txt"
Ratu : "Waktu eksekusi : 0.001548 detik"

```


testinput2.txt

```
SELAMAT DATANG DI 24 SOLVER
"Takjublah kalian dengan kecerdasan sang ratu"

Ratu : "Pilihan cara memilih kartu yang akan diberikan kepadaku!"
1. Random
2. Input kartu
> 2

Masukan kartu :
A J Q K

Ratu : "Apakah ingin menyimpan solusi yang ku dapat nanti ke dalam file?"
1. Ya
2. Tidak
> 1
Masukan nama file :
testinput2

(1*12)*(13-11)
1*(12*(13-11))
((1*13)-11)*12
(1*(13-11))*12
1*((13-11)*12)
(12*1)*(13-11)
12*(1*(13-11))
12*((1*13)-11)
(12/1)*(13-11)
12/(1/(13-11))
12*(13-(1*11))
12*((13*1)-11)
12*((13/1)-11)
(12*(13-11))*1
12*(13-(11*1))
12*((13-11)*1)
(12*(13-11))/1
12*(13-(11/1))
12*((13-11)/1)
(13-11)*(12*1)
((13-11)*12)*1
(13-11)*(12/1)
((13-11)*12)/1
(13-11)*(1*12)
((13-11)*1)*12
(13-(11*1))*12
((13-11)/1)*12
(13-(11/1))*12
(13-11)/(1/12)
(13-(1*11))*12
((13*1)-11)*12
((13/1)-11)*12
Ratu : "Aku berhasil mendapat 32 solusi."
Ratu : "Solusi berhasil disimpan di file testinput2.txt"
Ratu : "Waktu eksekusi : 0.001484 detik"
```

testinput3.txt

```

SELAMAT DATANG DI 24 SOLVER
"Takjublah kalian dengan kecerdasan sang ratu"

Ratu : "Pilihan cara memilih kartu yang akan diberikan kepadaku!"
1. Random
2. Input kartu
> 2

Masukan kartu :
2 3 4 5 6
Ratu : "Pastikan hanya 4 kartu!"
Masukan kartu :
1 2 3 4
Ratu : "Terdapat kartu yang salah!"
Masukan kartu :
1 2 3 4 5 6
Ratu : "Pastikan hanya 4 kartu!"
Ratu : "Terdapat kartu yang salah!"
Masukan kartu :
A B C D
Ratu : "Terdapat kartu yang salah!"
Masukan kartu :
A 2 J 4

Ratu : "Apakah ingin menyimpan solusi yang ku dapat nanti ke dalam file?"
1. Ya
2. Tidak
> 1
Masukan nama file :
testinput3

((1+11)/2)*4
(1+11)/(2/4)
(1+11)*(4-2)
(1+11)*(4/2)
((1+11)*4)/2
(2*(11-1))+4
((11+1)/2)*4
(11+1)/(2/4)
((11-1)*2)+4
(11+1)*(4-2)
(11+1)*(4/2)
((11+1)*4)/2
4+(2*(11-1))
(4-2)*(11+1)
(4/2)*(11+1)
4/(2/(11+1))
(4-2)*(1+11)
4-(2*(1-11))
(4/2)*(1+11)
4/(2/(1+11))
4+((11-1)*2)
(4*(11+1))/2
4*((11+1)/2)
4-((1-11)*2)
(4*(1+11))/2
4*((1+11)/2)
Ratu : "Aku berhasil mendapat 26 solusi."
Ratu : "Solusi berhasil disimpan di file testinput3.txt"
Ratu : "Waktu eksekusi : 0.000922 detik"

```

testrandom1.txt

```

      SELAMAT DATANG DI 24 SOLVER
"Takjublah kalian dengan kecerdasan sang ratu"

Ratu : "Pilihan cara memilih kartu yang akan diberikan kepadaku!"
1. Random
2. Input kartu
> 1

Mendapatkan angka...
Kartu yang didapat : (A = 1, J = 11, Q = 12, K = 13)
2 6 7 8

Ratu : "Apakah ingin menyimpan solusi yang ku dapat nanti ke dalam file?"
1. Ya
2. Tidak
> 1
Masukan nama file :
testrandom1
((2-6)+7)*8
(2-(6-7))*8
((2+7)-6)*8
(2+(7-6))*8
(2*(7+8))-6
(2*(8+7))-6
((7-6)+2)*8
(7-(6-2))*8
((7+2)-6)*8
(7+(2-6))*8
((7+8)*2)-6
8*((7-6)+2)
8*(7-(6-2))
((8+7)*2)-6
8*(7+(2-6))
8*((7+2)-6)
8*(2+(7-6))
8*((2+7)-6)
8*((2-6)+7)
8*(2-(6-7))
Ratu : "Aku berhasil mendapat 20 solusi."
Ratu : "Solusi berhasil disimpan di file testrandom1.txt"
Ratu : "Waktu eksekusi : 0.000755 detik"

```

testrandom2 (no solution)

```

      SELAMAT DATANG DI 24 SOLVER
"Takjublah kalian dengan kecerdasan sang ratu"

Ratu : "Pilihan cara memilih kartu yang akan diberikan kepadaku!"
1. Random
2. Input kartu
> 1

Mendapatkan angka...
Kartu yang didapat : (A = 1, J = 11, Q = 12, K = 13)
6 2 2 2

Ratu : "Apakah ingin menyimpan solusi yang ku dapat nanti ke dalam file?"
1. Ya
2. Tidak
> 1
Masukan nama file :
testrandom2
Ratu : "Aku tidak dapat menemukan solusi."
Ratu : "Tidak melakukan penyimpanan file."
Ratu : "Waktu eksekusi : 0.000103 detik"

```

LAMPIRAN

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	✓	