IF2211 Strategi Algoritma

Tugas Kecil 1

Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force

Oleh:

Reza Pahlevi Ubaidillah
13521165

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2023

Daftar Isi

Algoritma Brute Force

Brute force adalah cara naif dalam penyelesaian suatu permasalahan, yaitu dengan mencoba semua kemungkinan yang ada. Semua permasalahan dapat diselesaikan dengan pendekatan brute force, konsekuensi yang harus diterima adalah waktu. Sebuah komputer mungkin akan menyelesaikan permasalahan dengan jutaan kemungkinan dalam waktu singkat, tapi bagaimana dengan desiliunan (jumlah nol 33) kemungkinan? Tentu hal itu akan memakan waktu yang sangat lama atau bahkan tidak akan pernah terselesaikan dalam masa hidup komputer.

Dalam menyelesaikan tugas kecil ini, penulis membuat sebuah program sederhana dalam bahasa C++ melalui pendekatan *brute force* dengan langkah langkah sebagai berikut:

1. Periksa kasus trivial dari permainan. Kasus trivial adalah kasus di mana keempat kartu ketika dijumlahkan atau ketika dikalikan akan menghasilkan 24, yang berarti kita tidak perlu mengecek setiap permutasi kartu untuk kasus trivial. Misal, 4x3x2x1=24, 1x2x3x4=24 tidak perlu diperiksa.
2. Bangkitkan semua permutasi yang mungkin dari keempat kartu.
3. Kemudian periksa permutasi kartu dengan kombinasi operasi yang sudah terdaftar, apabila hasilnya 24 maka tuliskan ke layar.
4. Ulangi langkah 3 hingga semua permutasi sudah dicoba.

Source Code Program

Program terdiri atas 4 file yang ditulis dalam bahasa C++ (1 program utama dan 3 program modul pembantu). Program utama (main.cpp) berguna sebagai entrypoint di mana pengguna dapat menggunakan program ini.

```cpp
#include <iostream>
#include <string>
#include "solver.h"

int main(){
    char str[101];
    int input[4];
    int count;
    char mode[101];
    char save[101];

    printf("Pilih mode (m/r): ");
    fgets(mode, 100, stdin);
    if(mode[0]=='m' && strlen(mode)==2){
        printf("Masukkan 4 buah kartu: ");
        fgets(str, 100, stdin); // a op b op c op d
        append_to_ftext(str);
        append_to_ftext("\n");
    }
    else if(mode[0]=='r' && strlen(mode)==2){
        strcpy(str, generate_random().c_str());
        strcat(str, " ");
        printf("4 Kartumu: %s", str);
        append_to_ftext(str);
        append_to_ftext("\n");
    }
    else{
        printf("Masukan tidak valid!");
        return 0;
    }

    printf("\n");
```

Modul bantuan yang pertama adalah generator.cpp, berguna untuk membangkitkan semua permutasi yang ada.

```cpp
#include "generator.h"

void swap (int *arr, int a, int b){
    int temp;

    temp = arr[a];
    arr[a] = arr[b];
    arr[b] = temp;
}

void copy (int *dest, int src[], int size){
    int i;
    for(i = 0; i<size; i++){
        dest[i] = src[i];
    }
}

std::vector<int*> perm;

bool is_unique (int *arr){
    if(perm.empty()) return true;

    bool found;
    int size = perm.size();
    for(int i = 0; i<size; i++){
        found = true;
        for(int j = 0; j<4; j++){
            if(arr[j] != perm.at(i)[j]) found = false;
        }
        if(found) return false;
    }
    return true;
}

```

```cpp

 void generate_permutations(int *arr, int l, int r){
     int i;
     int temp;

     if (l==r){
         if(is_unique(arr)){
             int *result = (int*)malloc(sizeof(arr)*4);
             copy(result, arr, 4);
             perm.push_back(result);
         }
     }else {
         for(i=l; i<=r; i++){
             swap(arr, l, i);
             generate_permutations(arr, l+1, r);
             swap(arr, l, i);
         }
     }
 }

 std::vector<int*> get_permutations(int *arr){
     generate_permutations(arr, 0, 3);
     return perm;
 }
```

```cpp
#ifndef GENERATOR_H
#define GENERATOR_H

#include <iostream>
#include <vector>

int* generate_permutations(int arr[], int size);
std::vector<int*> get_permutations(int *arr);

#endif
```

Modul bantuan kedua adalah solver.cpp, di sini algoritma *brute force* berjalan. Ekspresi akan terbentuk dari pasangan operasi yang mungkin sesuai dengan permutasi yang diberikan, kemudian akan dihitung hasil akhirnya, jika 24 maka akan dicetak ke layar.

```h
#ifndef SOLVER_H
#define SOLVER_H

#include <cmath>
#include <iostream>
#include "utils.h"
#include "generator.h"

extern int count;

void solve_trivial(int arr[4]);
void solve(int arr[4]);
int get_count();


#endif
```

```cpp
#include "solver.h"

int count = 0;

void solve_trivial(int arr[4]){
    int i;
    int sum = 0;

    // Trivial a + b + c + d
    for(i=0; i<4; i++){
        sum += arr[i];
    }
    if(sum==24){
        print_expr(arr, "", "+", "+", "+", "");
        count++;
    }
    sum = 0;

    // Trivial a * b * c * d
    for(i=0; i<4; i++){
        sum *= arr[i];
    }
    if(sum==24){
        print_expr(arr, "", "*", "*", "*", "");
        count++;
    }
}
```

```cpp

void solve(int arr[4]){
    std::vector<int*> perms = get_permutations(arr);

    int *perm;
    do{
        perm = perms.back();
        // 2+; 1-
        if(perm[0] + perm[1] + perm[2] - perm[3] == 24) {print_expr(perm, "", "+", "+", "-", "");count++;}
        // 2+; 1*
        if(perm[0] * perm[1] + perm[2] + perm[3] == 24) {print_expr(perm, "", "*", "+", "+", "");count++;}
        if(perm[0] * (perm[1] + perm[2]) + perm[3] == 24) {print_expr(perm, "", "*(", "+", ")+", "");count++;}
        if(perm[0] * (perm[1] + perm[2] + perm[3]) == 24) {print_expr(perm, "", "*(", "+", "+", ")");count++;}
        // 1+; 2*
        if(perm[0] * perm[1] * perm[2] + perm[3] == 24) {print_expr(perm, "(", "*", "*", ")+", "");count++;}
        if(perm[0] * perm[1] * (perm[2] + perm[3]) == 24) {print_expr(perm, "(", "*", "*(", "+", "))");count++;}
        if(perm[0] * perm[1] + perm[2] * perm[3] == 24) {print_expr(perm, "(", "*", ")+(", "*", ")");count++;}
        // 1-; 2*
        if(perm[0] * perm[1] * perm[2] - perm[3] == 24) {print_expr(perm, "(", "*", "*", ")-", "");count++;}
        if(perm[0] * perm[1] * (perm[2] - perm[3]) == 24) {print_expr(perm, "(", "*", "*(", "-", "))");count++;}
        if(perm[0] * perm[1] - perm[2] * perm[3] == 24) {print_expr(perm, "(", "*", ")-(", "*", ")");count++;}
        // 1+; 1-; 1*
        if(perm[0] * perm[1] + perm[2] - perm[3] == 24) {print_expr(perm, "", "*", "+", "-", "");count++;}
        if(perm[0] * (perm[1] + perm[2]) - perm[3] == 24) {print_expr(perm, "", "*(", "+", ")-", "");count++;}
        if(perm[0] * (perm[1] - perm[2]) + perm[3] == 24) {print_expr(perm, "", "*(", "-", ")+", "");count++;}
        if(perm[0] * (perm[1] + perm[2] - perm[3]) == 24) {print_expr(perm, "", "*(", "+", "-", ")");count++;}
        if(perm[0] * perm[1] - (perm[2] + perm[3]) == 24) {print_expr(perm, "", "*", "-(", "+", ")");count++;}
        // 1+; 1*; 1/
        if(perm[0] * perm[1] == (perm[2]-24) * perm[3]) {print_expr(perm, "(", "*", "/", ")+", "");count++;}
        if(((perm[0] * perm[1]) + perm[2]) == 24 * perm[3]) {print_expr(perm, "((", "*", ")+", ")/", "");count++;}
        if((perm[0] + perm[1]) * perm[2] == 24 * perm[3]) {print_expr(perm, "((", "+", ")*", ")/", "");count++;}
        if(perm[0] * perm[1] == 24 * (perm[2] + perm[3])) {print_expr(perm, "(", "*", ")/(", "+", ")");count++;}
        // 1-; 1*; 1/
        if(perm[0] * perm[1] == (perm[2]+24) * perm[3]) {print_expr(perm, "(", "*", "/", ")-", "");count++;}
        if(((perm[0] * perm[1]) - perm[2]) == 24 * perm[3]) {print_expr(perm, "((", "*", ")-", ")/", "");count++;}
        if((perm[0] - perm[1]) * perm[2] == 24 * perm[3]) {print_expr(perm, "((", "-", ")*", ")/", "");count++;}
        if(perm[0] * perm[1] == 24 * (perm[2] - perm[3])) {print_expr(perm, "(", "*", ")/(", "-", ")");count++;}
        // 2*; 1/
        if(perm[0] * perm[1] * perm[2] == 24 * perm[3]) {print_expr(perm, "", "*", "*", "/", "");count++;}
        if(perm[0] * perm[1] == 24 * perm[2] * perm[3]) {print_expr(perm, "", "*", "/(", "*", ")");count++;}
        // 1-; 2/
        if(perm[0] * perm[3] == 24 * (perm[1] * perm[3] - perm[2])) {print_expr(perm, "", "/(", "-", "/", ")");count++;}
        // 1*; 2/
        if(perm[0] * perm[1] == 24 * perm[2] * perm[3]) {print_expr(perm, "(", "*", "/", ")/", "");count++;}

        perms.pop_back();
    }
    while(!perms.empty());

    std::string prompt;

    if(count==0) prompt = "Tidak ada solusi\n";
    else prompt = "\n" + std::to_string(count) + " solusi ditemukan";

    std::cout<<prompt<<std::endl;

    append_to_ftext(prompt);
}

int get_count(){
    return count;
}
```

Modul pembantu terakhir adalah utils.cpp, berbagai kebutuhan spesifik ada dalam modul ini.

src > C utils.h > ...

```h
#ifndef UTILS_H
#define UTILS_H

#include <cstring>
#include <cmath>
#include <string>
#include <iostream>
#include <cstdlib>
#include <time.h>
#include <fstream>
#include <chrono>

bool is_valid(char* str);
void parse(char* str, int* res);
std::string generate_random();
void print_expr(int arr[4], std::string prefix, std::string op1, std::string op2, std::string op3, std::string suffix);
void append_to_ftext(std::string str);
void save_to_text(std::string filename);

#endif
```

```cpp
#include "utils.h"

bool is_valid(char* str){
    int len = strlen(str);
    if(len!=8) return false;

    int i;
    for(i = 1; i<7 ; i+=2){
        if(str[i]!=' ') return false;
    }
    char key[] = "A23456789JQK";
    char * pch;
    for(i = 0; i<8; i+=2){
        pch = strchr(key, str[i]);
        if(pch==NULL) return false;
    }

    return true;
}

void parse(char* str, int* res){
    int i;
    int j = 0;
    char key[] = "A23456789JQK";
    char * pch;
    for(i = 0; i<8; i+=2){
        pch = strchr(key, str[i]);
        res[j] = pch - key + 1;
        j++;
    }
}
```

```
32
33  std::string generate_random(){
34      char key[] = "A23456789JQK";
35      std::string res;
36
37      int random;
38      int i;
39      srand(time(NULL));
40      for(i = 0; i<4; i++){
41          random = rand() % 12;
42          res.push_back(key[random]);
43          res.push_back(' ');
44      }
45      res.pop_back();
46
47      return res;
48  }
49
50  std::string ftext = "";
51
52  void print_expr(int arr[4], std::string prefix, std::string op1, std::string op2, std::string op3, std::string suffix){
53      std::string query("\n"+prefix+std::to_string(arr[0])+op1+std::to_string(arr[1])+op2+std::to_string(arr[2])+op3+std::to_string(arr[3])+suffix);
54      std::cout<<query;
55      ftext.append(query);
56  }
57
58  void append_to_ftext(std::string str){
59      ftext.append(str);
60  }
```

```
61
62  void save_to_text(std::string filename){
63      std::ofstream file(filename.c_str(), std::ofstream::out);
64
65      file<<ftext;
66
67      file.close();
68  }
```

Tangkapan Layar

Persoalan 1

```
PS D:\Kuliah\Semester 4\Stima\Tucil\1\src> .\a.exe
Pilih mode (m/r): r
4 Kartumu: A A 6 K

6*(1+1)+12
1 solusi ditemukan
Execution duration: 0.485 ms
Apakah kamu mau menyimpannya ke dalam file? (y)y
Simpan sebagai: test1

File berhasil disimpan
```

Persoalan 2

```
PS D:\Kuliah\Semester 4\Stima\Tucil\1\src> .\a.exe
Pilih mode (m/r): r
4 Kartumu: Q 4 J 3
Tidak ada solusi

Execution duration: 0.557 ms
Apakah kamu mau menyimpannya ke dalam file? (y)y
Simpan sebagai: test2

File berhasil disimpan
```

Persoalan 3

```
PS D:\Kuliah\Semester 4\Stima\Tucil\1\src> .\a.exe
Pilih mode (m/r): r
4 Kartumu: 8 6 K J

10+8+12-6
10+12+8-6
((10+6)*12)/8
12+10+8-6
12+8+10-6
(12*8)/(10-6)
6*(10-8)+12
((6+10)*12)/8
(6*8)/(12-10)
8+10+12-6
8+12+10-6
(8*12)/(10-6)
(8*6)/(12-10)
13 solusi ditemukan
Execution duration: 2.545 ms
Apakah kamu mau menyimpannya ke dalam file? (y)y
Simpan sebagai: test3

File berhasil disimpan
```

Persoalan 4

```
PS D:\Kuliah\Semester 4\Stima\Tucil\1\src> .\a.exe
Pilih mode (m/r): m
Masukkan 4 buah kartu: A A 8 3


3*8+1-1
3*(8+1-1)
3*8*1/1
3*8/(1*1)
(3*8/1)/1
3*(1+8-1)
3*1*8/1
8*3+1-1
8*(3+1-1)
8*3*1/1
8*3/(1*1)
(8*3/1)/1
8*(1+3-1)
8*1*3/1
1*3*8/1
1*8*3/1
16 solusi ditemukan
Execution duration: 3.753 ms
Apakah kamu mau menyimpannya ke dalam file? (y)y
Simpan sebagai: test4

File berhasil disimpan
```

Persoalan 5

```
PS D:\Kuliah\Semester 4\Stima\Tucil\1\src> .\a.exe
Pilih mode (m/r): m
Masukkan 4 buah kartu: 2 5 7 9


7*5-(9+2)
7*5-(2+9)
5*7-(9+2)
5*7-(2+9)
4 solusi ditemukan
Execution duration: 1.268 ms
Apakah kamu mau menyimpannya ke dalam file? (y)y
Simpan sebagai: test5

File berhasil disimpan
```

Persoalan 6

```
PS D:\Kuliah\Semester 4\Stima\Tucil\1\src> .\a.exe
Pilih mode (m/r): m
Masukkan 4 buah kartu: Q Q Q Q

Tidak ada solusi

Execution duration: 0.285 ms
Apakah kamu mau menyimpannya ke dalam file? (y)y
Simpan sebagai: test6

File berhasil disimpan
```

Tautan Kode Program

Untuk mencoba kode sumber program yang lebih lengkap dan lebih mudah dibaca, silakan askses repository GitHub berikut:

https://github.com/obediqbal/24Solver

Tabel Pengecekan Spesifikasi

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	✓	