Laporan Tugas Kecil 1 IF2211 Strategi Algoritma

**Penyelesaian Permainan Kartu 24 dengan Algoritma Brute-force**

Ditujukan untuk memenuhi tugas kecil I mata kuliah IF2211 Strategi Algoritma Semester II tahun akademik 2022/2023

Logo

Description automatically generated

Disusun oleh:

Muhammad Zaki Amanullah – 13521146

**Program Studi Teknik Informatika**

**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

**2023**

1. **Algoritma Brute-force**

Algoritma brute-force adalah strategi pemecahan masalah dengan mencoba semua kemungkinan dari permasalahan. Karena “approach” dari algoritma ini yang relatif sederhana, yaitu dengan mencoba segala kombinasi kemungkinan, penyelesaian masalah dengan algoritma ini relatif “straight-forward”. Dalam tugas ini, saya akan mengimplementasikan algoritma brute-force untuk mencari penyelesaian atau solusi dari puzzle atau teka-teki 24.

Dalam teka-teki ini, diberikan empat buah angka, yang harus kita ubah menjadi angka 24 dengan menggunakan operasi penjumlahan, pengurangan, perkalian, atau pembagian. Dalam, masalah ini, angka pada teka-teki 24 yang digunakan diambil dari satu dek kartu yang berisikan A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K. Program yang saya buat, akan menghasilkan semua solusi yang mungkin menghasilkan angka 24 dengan menggunakan algoritma brute-force.

Alur yang dilalui oleh program yang dibuat adalah sebagai berikut.

1. Program menerima 4 buah angka yang dapat diambil dari input user atau secara random.
2. Program mencari kombinasi dari urutan angka tersebut dan kombinasi dari operator-operator yang mungkin serta semua kemungkinan *grouping* dan mengevaluasi hasilnya.
3. Apabila hasil yang didapat adalah 24, program akan memasukkan string berisikan salah satu solusi ke dalam vector.

Secara teori karena terdapat 4 buah angka yang dianggap berbeda, kita dapat mendapatkan 4! buah susunan angka yang berbeda. Operator dapat disusun menjadi 4.4.4 buah susunan yang berbeda karena setiap tempat menempatkan operator kita memiliki empat buah pilihan. Terdapat lima kemungkinan grouping yang mungkin sebagai berikut.

1. (( a op b ) op c ) op d
2. ( a op ( b op c )) op d
3. a op (( b op c ) op d )
4. a op ( b op ( c op d ))
5. ( a op b ) op ( c op d)

Dapat diperhatikan, bahwa dengan mempertimbangkan semua kemungkinan, program akan mengevaluasi sebanyak 7680 buah kemungkinan yang relatif mudah untuk diselesaikan oleh komputer. Oleh karena itu, permasalahan ini cocok digunakan untuk algoritma brute-force. Berikut adalah approach saya dalam menyelesaikan masalah ini.

1. Mengubah string menjadi operasi

A screenshot of a computer

Description automatically generated with medium confidence

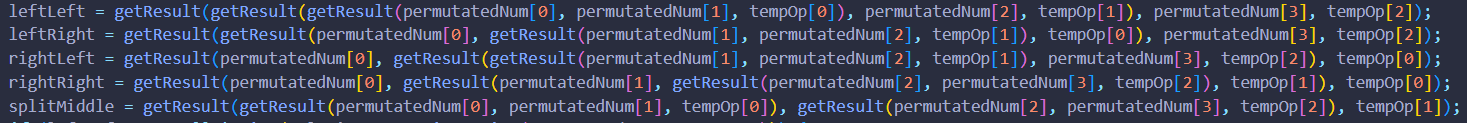
Karena program akan mendapatkan input berupa string, diperlukan sebuah fungsi yang melakukan operasi antara dua angka. Dapat dilihat pada fungsi getResult terdapat suatu metode untuk menghindari terjadinya error yang akan terjadi apabila terdapat pembagian dengan angka 0. Apabila terdapat kasus seperti itu, dikembalikan nilai 99999 yang dipilih karena nilai tersebut lebih tinggi dari semua angka yang akan dicapai pada permasalahan ini (28561). Apabila terdapat satu operasi yang menghasilkan 99999 maka operasi selanjutnya akan selalu menghasilkan 99999 dan tidak mencapai 24.

1. Mencari semua kombinasi

A picture containing text

Description automatically generated

Untuk mendapatkan semua kemungkinan susunan operator dan susunan angka yang mungkin, digunakan nested for loop sebanyak 7 kali.



Text

Description automatically generated

Setelah didapatkan semua kombinasi dari angka dan operator, kita lakukan operasi berdasarkan semua kemungkinan grouping yang ada. Kemudian, kita akan evaluasi apakah hasil yang didapat adalah 24 dan apakah solusi yang didapat sudah terdapat pada vector of string yang digunakan untuk menyimpan solusi. Apabila memenuhi kedua hal tersebut, maka solusi akan dimasukkan ke dalam vector.

1. Mendapatkan string dari solusi

A picture containing text

Description automatically generated

Karena grouping yang memungkinkan hanya terdapat 5 buah, dapat dibuat sebuah fungsi yang akan mengembalikan sebuah string yang sesuai dengan solusi yang ada. Format yang dimasukkan merupakan angka dari 1—5 yang direpresentasikan oleh grouping yang telah dibahas sebelumnya.

1. Menghindari hasil kembar

Text

Description automatically generated

Karena terdapat kemungkinan apabila terdapat solusi yang sama maka dibuatlah fungsi exist yang mengembalikan true apabila vector of string yang digunakan untuk menyimpan solusi sudah memiliki solusi tersebut dan false apabila tidak.

1. **Source Code Program**

Semua program yang dibuat diletakkan di dalam file main.cpp yang berisi semua fungsionalitas dari pembacaan input, pencarian angka secara random, algoritma brute force, dan penulisan ke file txt.

1. Pembacaan input.

Angka yang didapat dapat berasal dari input pengguna atau berasal dari program. Untuk mendapatkan angka dari pengguna, pengguna diharapkan untuk menuliskan 4 buah nomor kartu yang valid yang dipisahkan oleh white space dan diakhiri oleh enter.

Text

Description automatically generated

Untuk memudahkan parsing kartu, dibuatlah fungsi diatas yang akan menghapus leading dan trailing white space yang mungkin akan ada saat pengguna memasukkan input.

Text

Description automatically generated

Untuk mengetahui apakah kartu yang dimasukkan merupakan kartu yang valid, kita lakukan pengecekan satu per satu akan setiap ‘kata’ yang dimasukkan pengguna. Apabila setelah parsing terdapat kurang dari atau lebih dari 4 buah kartu, maka input tidak valid. Apabila terdapat sebuah kartu yang tidak terdapat dalam dek, maka masukkan akan dianggap tidak valid dan pengguna akan diminta untuk memasukkan kembali 4 buah kartu yang valid.

Text

Description automatically generated

Setelah semua input dipastikan sudah valid, akan dijalankan fungsi berikut yang akan mengubah variabel cardValues menjadi array yang berisikan angka yang dimasukkan pengguna.

Text

Description automatically generated

1. Mendapatkan angka secara acak.

Text

Description automatically generated

Untuk mendapatkan angka secara acak, digunakan fungsi dari standard library c++ srand dan rand. srand merupakan fungsi yang digunakan untuk mendapatkan seed untuk memastikan bahwa angka acak yang didapatkan berbeda setiap waktunya. Lalu untuk mendapatkan angka acaknya digunakan rand yang dilanjutkan dengan operasi modulo untuk memastikan bahwa angka acak yang didapat dalam kisaran 0-13.

1. Mendapatkan pilihan dari user.
2. Algoritma brute-force

Text

Description automatically generated

Chart

Description automatically generated with low confidence

Terdapat penggunaan library chrono untuk mendapatkan waktu eksekusi program dalam microsecond lalu dimasukkan ke dalam solusi untuk selanjutnya dimasukkan ke dalam file txt dengan mudah.

1. Penulisan ke file txt

Text

Description automatically generated

Program dapat menaruh hasil yang didapat ke dalam text file. Apabila user memasukkan ‘Y’ atau ‘y’ maka pengguna akan diminta untuk memasukkan nama file yang valid, yaitu nama file yang belum ada. Apabila nama file sudah ada, maka pengguna akan diminta untuk memasukkan kembali nama file.

Text

Description automatically generated

Untuk mengetahui apakah sebuah file sudah ada, dibuat fungsi isFileExist yang memanfaatkan library fstream dari c++. Untuk menulis hasil ke file text digunakan kelas ofstream dari library fstream.

1. **Penggunaan program dan contoh input/output**

Contoh input/output:

1. Kartu masukkan
   1. Dari pengguna
      * Input valid

Text

Description automatically generated

* + - Input tidak valid (jumlah kartu tidak sesuai)

Text

Description automatically generated

* + - Input tidak valid (kartu tidak sesuai)

Text

Description automatically generated

* 1. Secara acak

Shape

Description automatically generated with low confidence

1. File txt

Text

Description automatically generated with medium confidence

File txt yang dihasilkan:

A picture containing text

Description automatically generated

1. Full program flow
   1. Tidak ada solusi

Text

Description automatically generated

* 1. Solusi beragam

Background pattern

Description automatically generated with medium confidence

File txt yang dihasilkan:

A picture containing background pattern

Description automatically generated

* Semua angka sama

Text

Description automatically generated

* Randomize

Text

Description automatically generated

Text

Description automatically generated

1. **Link Repository**

<https://github.com/zakia215/Tucil1_13521146>

1. **Tabel**

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program berhasil dikompilasi tanpa kesalahan | ✓ |  |
| 1. Program berhasil *running* | ✓ |  |
| 1. Program dapat membaca input / generate sendiri dan memberikan luaran | ✓ |  |
| 1. Solusi yang diberikan program memenuhi (berhasil mencapai 24) | ✓ |  |
| 1. Program dapat menyimpan solusi dalam file teks | ✓ |  |