

TUGAS KECIL 1
IF2211 – Strategi Algoritma

Penyelesaian Permainan Kartu 24
dengan Algoritma Brute Force



Disusun oleh

Frankie Huang 13521092

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

T.A. 2022/2023

A. Algoritma Brute Force

Algoritma *brute force* adalah pendekatan paling *straightforward* untuk memecahkan suatu persoalan. Pada *source file*, algoritma brute force digunakan untuk mencari semua jawaban yang mungkin dan untuk melakukan pencarian untuk mengubah integer menjadi string dan sebaliknya.

Misalkan terdapat 4 buah bilangan: *a*, *b*, *c*, dan *d*; dan 4 buah operasi yang dapat dilakukan: "+", "-", "*", dan "/". Untuk operasi 4 buah bilangan, terdapat 5 jenis urutan pengoperasian yang dapat dilakukan, yaitu:

1. ((*a op1 b*) *op2 c*) *op3 d*
2. (*a op1 (b op2 c)*) *op3 d*
3. (*a op1 b*) *op2 (c op3 d)*
4. *a op1 ((b op2 c) op3 d)*
5. *a op1 (b op2 (c op3 d))*

Namun, urutan pengoperasian 1 dan 5 serupa karena keempat bilangan akan dipermutasikan, sehingga salah satu urutan pengoperasian dapat diabaikan. Kemudian, lakukan nested loop 3 kali, masing-masing untuk tiap *op*, sehingga untuk keempat bilangan yang berbeda, terdapat maksimal $4! \cdot 3^4 \cdot 5$ buah urutan pengoperasian yang akan diperiksa oleh program.

Untuk algoritma *brute force* yang digunakan untuk melakukan pencarian, cara yang digunakan cukup mudah dimengerti. Pencarian dilakukan secara linear hingga ditemukan dan jika tidak ditemukan, maka ada kesalahan pada input pengguna yang akan mengakibatkan pada berhenti berjalannya program.

B. Implementasi pada Bahasa Pemrograman C++

Pada *source file*, terdapat 4 buah variabel global dan 3 buah struktur data global yang dijelaskan pada tabel di bawah ini

Nama	Fungsi
<code>char input[16]</code>	Menyimpan input pengguna dalam bentuk <i>array of character</i>
<code>int integer[4]</code>	Menyimpan keempat bilangan dalam bentuk <i>integer</i>
<code>int total</code>	Menyimpan total jawaban yang mungkin didapatkan
<code>vector<float> numbers</code>	Menyimpan keempat bilangan dalam bentuk <i>float</i>
<code>vector<string> answers</code>	Menyimpan jawaban yang didapatkan

<code>set<vector<float>> permutations</code>	Mengecek apakah permutasi sudah pernah ditemukan sebelumnya
--	---

Selain itu, terdapat juga fungsi-fungsi yang terdapat pada *source file*, yaitu

Nama	Fungsi
<code>float convert(char c)</code>	Mengubah <i>input</i> dalam bentuk <i>char</i> menjadi <i>float</i> serta mengubah karakter “*” menjadi angka random
<code>void readInput()</code>	Menerima input pengguna dan melakukan pengecekan input pengguna
<code>float operate(float a, float b, int op)</code>	Melakukan operasi antara <i>float</i> a dan <i>float</i> b sesuai dengan nilai <i>op</i>
<code>string insertOperator(int op)</code>	Mengembalikan string sesuai dengan nilai <i>op</i>
<code>void findResult(int i, int j, int k, int type)</code>	Menyusun urutan pengoperasian yang mungkin dan menyimpannya pada <i>answers</i>
<code>void swap(int a, int b)</code>	Melakukan perubahan nilai pada indeks <i>a</i> dan <i>b</i> pada <i>numbers</i>
<code>void findNumbers(int n)</code>	Melakukan algoritma <i>backtracking</i> untuk mencari seluruh permutasi bilangan
<code>int main()</code>	Memanggil fungsi lain serta menghitung waktu yang dibutuhkan dan penyimpanan jawaban pada file

C. Eksperimen

1. Kasus 4 buah bilangan (1 2 3 4)

```

((4 * 1) * 2) * 3
(4 * (1 * 2)) * 3
(4 * 1) * (2 * 3)
4 * (1 * (2 * 3))
4 * ((1 * 2) * 3)
((4 / 1) * 2) * 3
(4 / 1) * (2 * 3)
(4 / (1 / 2)) * 3
4 / (1 / (2 * 3))
4 / ((1 / 2) / 3)

Elapsed time: 370 microseconds.

Would you like to save the output to a file? (Y/n)
Insert file name: TC_1.txt

```

2. Kasus 4 buah bilangan acak (* * * *)

```

Please input 4 valid cards (or * to randomize one card) seperated by space.
* * * *

4 solutions found for numbers 11, 9, 2, and 1.

((2 + 1) * 11) - 9
((1 + 2) * 11) - 9
(11 * (2 + 1)) - 9
(11 * (1 + 2)) - 9

Elapsed time: 81 microseconds.

Would you like to save the output to a file? (Y/n)y
Insert file name: TC_2.txt

```

3. Kasus karakter A, J, Q, dan K (a j q k)

```

Please input 4 valid cards (or * to randomize one card) seperated by space.
a j q k

32 solutions found for numbers 13, 1, 11, and 12.

(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

Elapsed time: 256 microseconds.

Would you like to save the output to a file? (Y/n)
Insert file name: TC_3.txt

```

4. Kasus 2 bilangan serupa (8 4 7 7)

```

Please input 4 valid cards (or * to randomize one card) seperated by space.
8 4 7 7

10 solutions found for numbers 7, 7, 4, and 8.

8 * (4 - (7 / 7))
4 * ((7 - 8) + 7)
4 * (7 - (8 - 7))
(4 - (7 / 7)) * 8
4 * (7 + (7 - 8))
4 * ((7 + 7) - 8)
((7 - 8) + 7) * 4
(7 - (8 - 7)) * 4
((7 + 7) - 8) * 4
(7 + (7 - 8)) * 4

Elapsed time: 165 microseconds.

Would you like to save the output to a file? (Y/n)
Insert file name: TC_4.txt

```

5. Kasus 2 bilangan acak (4 7 * *)

```
Please input 4 valid cards (or * to randomize one card) seperated by space.
4 7 * *

18 solutions found for numbers 3, 3, 7, and 4.

((4 + 7) - 3) * 3
(4 + (7 - 3)) * 3
4 * (7 - (3 / 3))
((4 - 3) + 7) * 3
(4 - (3 - 7)) * 3
((7 + 4) - 3) * 3
(7 + (4 - 3)) * 3
((7 - 3) + 4) * 3
(7 - (3 - 4)) * 3
(7 - (3 / 3)) * 4
3 * (7 + (4 - 3))
3 * ((7 + 4) - 3)
3 * ((7 - 3) + 4)
3 * (7 - (3 - 4))
3 * (4 + (7 - 3))
3 * ((4 + 7) - 3)
3 * ((4 - 3) + 7)
3 * (4 - (3 - 7))

Elapsed time: 159 microseconds.

Would you like to save the output to a file? (Y/n)
Insert file name: TC_5.txt
```

6. Kasus karakter A dan nilai 1 (a j 1 a)

```
Please input 4 valid cards (or * to randomize one card) seperated by space.
a j 1 a

4 solutions found for numbers 11, 1, 1, and 1.

(1 + 11) * (1 + 1)
(1 + 1) * (11 + 1)
(1 + 1) * (1 + 11)
(11 + 1) * (1 + 1)

Elapsed time: 45 microseconds.

Would you like to save the output to a file? (Y/n)
Insert file name: TC_6.txt
```

7. Kasus angka tidak valid (19 a a a)

```
Please input 4 valid cards (or * to randomize one card) seperated by space.
19 a a a

Invalid input, please try again!
```

8. Kasus angka tidak valid (0 a a a)

```
Please input 4 valid cards (or * to randomize one card) seperated by space.  
0 a a a  
Invalid input, please try again!
```

9. Kasus karakter tidak valid (b a a a)

```
Please input 4 valid cards (or * to randomize one card) seperated by space.  
b a a a  
Invalid input, please try again!
```

10. Kasus input kurang dari 4 (1 2 3)

```
Please input 4 valid cards (or * to randomize one card) seperated by space.  
1 2 3  
Invalid input, please try again!
```

11. Kasus input lebih dari 5 (1 2 3 4 5)

```
Please input 4 valid cards (or * to randomize one card) seperated by space.  
1 2 3 4 5  
Invalid input, please try again!
```

D. Pranala

Pranala spesifikasi tugas :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/Tucil1-Stima-2023.pdf>

Pranala repository : https://github.com/frankiehuangg/Tucil1_13521092