

The Application of Brute Force Algorithm to Determine The Isomers of an Alkane Compound

Juan Christopher Santoso - 13521116
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): juan.csantoso@gmail.com

Abstract—Alkane is one of the basic hydrocarbon compounds in constructing organic elements in the whole universe. Like the other type of hydrocarbon compounds, alkane might have isomers depending on the amount of carbon and hydrogen it contains. Isomers, on the other hand, can be simply described as alternative structural arrangements of a compound's components. Isomers of an alkane compound can be determined by trying every possible combination to construct an alkane of the same molecular formula. Thus, the brute force algorithm can be one of many ways to solve this specific problem. The brute force algorithm is implemented using Java programming language.

Keywords—alkane, hydrocarbon, isomers, brute force, algorithm

I. INTRODUCTION

Matters are physical substances in creating everything people have known so far. Discussing deeper about matters, matters come in different forms, starting from normal matters and anti-matters, that people have been researching, or the dark matters, the ones that still hold a big question mark for humanity. Normal matters are the ones people used to discuss when it comes to physical substances and chemical reactions because this type of matter is the one which can be relatively easy to be observed.

Normal matters can be divided into two major categories, organic and non-organic. Organic elements, or can also be said as organic compounds, come with many variations with hydrocarbon compounds as their base. Hydrocarbon is an organic compound, consisting of Hydrogen and Carbon. Hydrocarbon itself comes into several types based on the types of carbon bonds it contains, which are alkane, alkene, and alkyne. Alkane is a type of hydrocarbon compound with a generic type of carbon bond, which is a single bond. On the other hand, if a hydrocarbon contains a double bond, it is called alkene, if a triple bond, alkyne.

Every compound categorized as a hydrocarbon compound, can be existed with several types of structural arrangement. Scientists called this alternative form of a hydrocarbon compound an isomer. Different type of hydrocarbon has a different type of isomer.

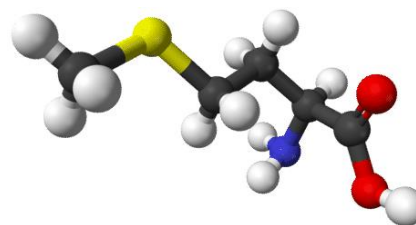


Image 1.1 Visualization of an Organic Compound in Ball-and-stick Model

(Source:

https://chem.libretexts.org/Bookshelves/Introductory_Chemistry/Introductory_Chemistry_%28CK-12%29/25%3A_Organic_Chemistry/25.01%3A_Organic_Chemistry)

The amount of isomer grows exponentially to the amount of carbon and hydrogen atoms inside the alkane itself. The more atoms it contains, the more isomers can exist of that specific molecular formula. Therefore, to determine every possible combination, a brute force algorithm can be implemented.

II. THEORETICAL BASIS

A. Hydrocarbon Compound

Organic compounds are defined as covalently bonded compounds containing carbon and hydrogen [1]. Hydrocarbon itself is generally visualized as a chain-like compound. Therefore, based on the type of its “chain-like structures”, hydrocarbon can be divided into three categories [2]:

1. Aliphatic,
2. Alicyclic, and
3. Aromatic.

Aliphatic hydrocarbon can be defined as one with acyclic structures, or an open chain. Since hydrocarbon consists of carbon atoms, the type of carbon bonds affects the type of hydrocarbon. Based on the types of bonds it contains, aliphatic hydrocarbon can be categorized as [3]:

1. Alkanes, contains only single bonds
2. Alkenes, contains at least one double bond, and
3. Alkynes, contains at least one triple bond.

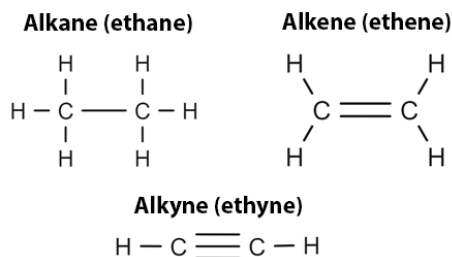


Image 2.1 Alkane, Alkene, and Alkyne

(Source: <https://homework.study.com/explanation/what-is-the-difference-between-alkanes-alkenes-and-alkynes.html>)

B. Alkane

Alkane is a type of hydrocarbon compound, where all the bonds between the carbon atoms are single bonds. Alkane has the molecular formula of $\text{C}_n\text{H}_{2n+2}$, with n is the amount of carbon atoms in the alkane compounds. Alkanes can be subdivided into three groups, which are [4]:

1. Linear straight-chain alkanes,
2. Branched Alkanes, and
3. Cycloalkanes.

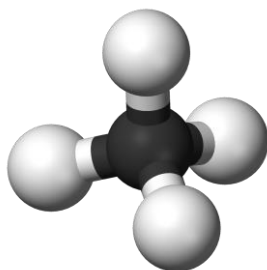


Image 2.2 Visualization of Methane, an Alkane Compound (CH_4)

(Source: <https://en.wikipedia.org/wiki/Hydrocarbon>)

Name of all alkane compounds end with an -ane. Here are the list of the names of alkane compounds, starting with one carbon atom up to ten carbon atoms:

Table 2.1 List of Alkane Compounds Name

Molecular Formula	Name
CH_4	Methane
C_2H_6	Ethane
C_3H_8	Propane
C_4H_{10}	Butane
C_5H_{12}	Pentane
C_6H_{14}	Hexane
C_7H_{16}	Heptane
C_8H_{18}	Octane
C_9H_{20}	Nonane
$\text{C}_{10}\text{H}_{22}$	Decane

C. Isomer

Isomer is the phenomenon in which more than one compounds share the same molecular formula but have different molecular structures [5]. The word isomer itself is derived from Greek words “isos” and “meros”, which mean “equal parts”.

There are two major types of isomerism, which are the structural isomerism and stereoisomerism. Structural isomerism is the type of isomer phenomenon where the structure of the chemical compound actually differs one to another. On the other hand, stereoisomerism where the chemical compound’s structure does not change, but seems to be different because of the observer’s stereotype.

Structural isomerism, can be divided deeper into several subcategories, which are:

1. Chain Isomerism,
2. Position Isomerism,
3. Functional Isomerism,
4. Metamerism,
5. Tautomerism, and
6. Ring-Chain Isomerism.

On the other hand, stereoisomerism is only divided into two, known as, Geometric Isomerism and Optical Isomerism.

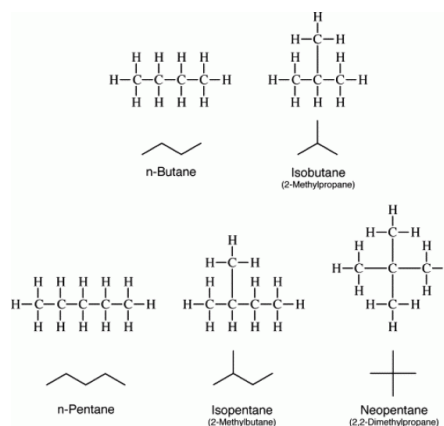


Image 2.3 Chain Isomerism, an example type of Structural Isomerism

(Source: <https://www.differencebetween.com/what-is-the-difference-between-chain-isomerism-and-position-isomerism/>)

D. Brute Force Algorithm

Brute Force Algorithm is a straightforward method of solving a problem that rely on sheer computing power [6]. Brute Force itself is derived from the words, brute and force. Brute has the meaning of “absence of reasoning or intelligence” while force means “coercion or compulsion”.

This type of algorithm is mainly related to problem-solving methods that do not need to consider effectiveness and efficiency. Trial-and-error with traversing every possible combination explains how this algorithm works. The sentence of “Just do it!” is the sole principle of implementing Brute Force Algorithm [7].

III. APPLICATION

The program implemented using Brute Force Algorithm will be developed using Java programming language. The structure of the alkane compound will be visualized by a matrix of integer, where the integer value of each tile of the matrix represents the carbon atom and the amount of hydrogen atoms it connects.

A. Input/Output

Before the program starts, the program will ask the user to input a certain amount of carbon atoms in the alkane. For example, if the user inputs "5", then it indicates the isomer search for alkane with 5 atoms of carbon. The program will automatically calculate the input as C_nH_{2n+2} formula, where the input of "5" will be registered as isomer search for C_5H_{12} .

The output of the program will be every possible combination for the specific alkane molecular structure. Displaying the combination of the molecular structure is optional, but the amount of combination will still be displayed either way.

B. Matrix

The matrix is the 'field' to represent the molecular structure of the alkane. The matrix used in the program is a matrix of integer, with the size of $[m \times n]$, with n is the amount of carbon atoms in the alkane compound and m is equal to $\lfloor n/2 \rfloor$.

The matrix has the range value of -1 to 3, with the default value of -1. The value of -1 represents that the tile is empty while the value of 0 to 3 indicates that the specific tile contains a carbon atom.

The matrix itself is contained as an attribute in a class, named Map, with the constructor and setter getter methods is displayed as below:

```
public class Map {
    private int[][] m;

    public Map(int row, int col){
        this.m = new int[row][col];
        for (int i = 0 ; i < row; i++){
            for (int j = 0; j < col; j++){
                m[i][j] = -1;
            }
        }
    }

    public int getRow(){
        return m.length;
    }

    public int getCol(){
        return m[0].length;
    }

    public void setIntToMap(int n, int r, int c){
```

```
        m[r][c] = n;
    }

    public int getIntFromMap(int r, int c){
        return m[r][c];
    }
}
```

C. Visualization of Alkane Compound

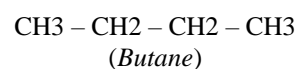
As mentioned in the previous point, the structure of the alkane compound will be represented using a matrix of integer. Each tile of the matrix represents whether a carbon compound is contained with a positive value of 0 to 3 as the indication. Each value can be explained as:

1. The value 0, indicates that the carbon in the specific tile is quaternary carbon, with no direct bond to hydrogen atoms,
2. The value 1, indicates that the carbon is a ternary carbon, with only one direct bond to a hydrogen atom,
3. The value 2, represents secondary carbon with two direct bonds to hydrogen atoms, and
4. The value 3, represent primary carbon, with three direct bonds to hydrogen atoms.

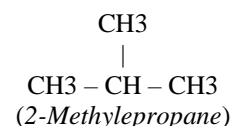
Tile with the value of -1 will not be displayed to the user. It will be displayed as a whitespace as it indicates that is empty.

Image 3.1 and 3.2 Isomer of C_4H_{10} Displayed in Matrix
(Source: Author's archive)

Alkane compounds, as it is displayed in Image 3.1 and 3.2 are both have chain isomers one to another. Both matrix in Image 3.1 and 3.2 represents the same C_4H_{10} , but in the different structure as in,



and



D. Container

Every isomer that has been validated will be stored inside a container. This container is contained as an attribute of a class named, IsomerContainer.

E. Looping Algorithm

In order to search all the isomers of an alkane, the program uses the brute force algorithm. Therefore, since the brute force algorithm is implemented, the program cannot be separated with loops and iteration.

There are two major loops in this program, the fixed loop, using only for loop, and the recursion, using a recursive function. Steps in the looping algorithm can be described as:

1. Iterate every row in the matrix.
2. For every row iteration, iterate every column inside that specific row.
3. For every column, call the recursive function to process every tile that has been found so far.

While the iteration process inside the recursive algorithm can be explained as:

1. The specific tile will be iterated 5 times, for every possible range of value, -1 to 3.
2. If the checked tile is not in first row neither in the first column, for every iteration, call the recursive function to do the recursion to the one tile to the left (one column before).
3. If the checked tile is not in the first row but in the first column, for every iteration, call the recursive function to do the recursion for the last column of the row above.
4. If the checked tile is in the first row and in the first column (on the top-left of the matrix), call the checking function to validate whether the map is an isomer for the alkane compound.

```
// Looping
for(int i = 0; i < row; i++){
    for (int j = 0; j < col; j++){
        recursiveLoop(ic, m, i, j, row-1, col-1);
        m.resetMap();
    }
}

public static void
recursiveLoop(IsomerContainer ic, Map m, int r,
int c, int maxRow, int maxCol){
    if (r == 0 && c == 0){
        // Special Case for Methane
        if (maxCol == 0 && maxRow == 0){
            m.setIntToMap(4, maxRow,
maxCol);
        }
        // Handle case for Small n
        if (r == maxRow/2){
            m.setIntToMap(3, r, c);
        }
        // Check if it is an Isomer
        if (m.isAnIsomer(maxCol+1)){
            // Prevent double output
            if
```

```
(!ic.isMapAlreadyInContainer(m)){
            ic.addMap(m);
            countIncrement();
        }
    }
}
// Rekurens
else {
    if (c > 0){
        for (int itr = -1; itr < 4;
itr++){
            m.setIntToMap(itr, r, c);
            recursiveLoop(ic, m, r, c-1, maxRow, maxCol);
        }
    }
    else {
        if (r == maxRow/2){
            m.setIntToMap(3, r, c);
        } else {
            m.setIntToMap(-1, r, c);
        }
        recursiveLoop(ic, m, r-1,
maxCol, maxRow, maxCol);
    }
}
}
}
```

F. Checking Functions

There are several checking functions implemented in the program in order to make sure that the structure found in the matrix is indeed an isomer of the specific alkane compound. Those checking functions can be divided into two main which are:

1. Amount Checking

Amount checking is needed to make sure that all the components in the matrix has the same amount of carbon and hydrogen atoms that the user has inputted. The algorithm to use this function is to traverse every row and every column in the matrix. While doing iteration, the program will count the amount of tile that is not empty and the total amount value of that these tiles contain. The amount of not empty tile indicates should be equal to the amount of the carbon atoms, while on the other hand, the total value these tiles contain should be equal to the hydrogen atoms that the alkane compound contains.

The details of this type of checking is displayed below:

```
public boolean isAnIsomer(int n){
    int carbonCount = 0;
```

```

int hydrogenCount = 0;
for(int i = 0; i < getRow(); i++){
    for(int j = 0; j < getCol();
j++){
        if (m[i][j] != -1){
            if (isValidCarbon(i,
j)){
                carbonCount++;
                hydrogenCount +=
m[i][j];
            }
            else {
                return false;
            }
        }
    }
}
return (carbonCount == n &&
hydrogenCount == (2*n+2));
}

```

2. Behavior Checking

Behavior checking is needed to make sure that all the value of the isomer solution has the same behavior as how a carbon should be inside an alkane compound. Those behavior of the carbon atoms can be described as:

1. Every primary carbon that has the position at the edge of the molecule should have bonds with 3 hydrogen atoms, making it have the value of 3.
2. Every secondary carbon, that has the position between two other carbons, should have bonds with 2 hydrogen atoms, making it have the value of 2.
3. Every ternary carbon only has the value of 1, since it only connects with one hydrogen atom.
4. Every quaternary carbon cannot have any bonds with any hydrogen atoms, making it only has the value of 0.

The algorithm to implement behavior checking will be displayed as:

```

public boolean isValidCarbon(int i, int
j){
    int count = 4;
    if (i != 0 && getIntFromMap(i-1, j)
!= -1){
        count--;
    }
    if (i != getRow()-1 &&
getIntFromMap(i+1, j) != -1){

```

```

        count--;
    }
    if (j != 0 && getIntFromMap(i, j-1)
!= -1){
        count--;
    }
    if (j != getCol()-1 &&
getIntFromMap(i, j+1) != -1){
        count--;
    }
    return (count == getIntFromMap(i,
j));
}

```

3. Double Appearance Checking

Double appearance checking is needed to make sure that no literal same structure will be displayed more than once in solutions container. For every isomer matrix inside the IsomerContainer will be checked before another isomer solution is added into the container. The logic behind this checking is to traverse all the row and column between 2 matrices. If it is found an unequal integer between those matrices, then this checking function will return 'false'. On the other hand, if after all iterations are finished and there is no unequal integer between the two matrices, then the same matrix has been contained in the container.

The detail of the implementation of this checking function is displayed as below:

```

// Class IsomerContainer
public void addMap(Map m){
    Map temp = new Map(m.getRow(),
m.getCol());
    temp.copyFrom(m);
    container.add(temp);
}

public boolean
isMapAlreadyInContainer(Map m){
    for (Map map : container){
        if (map.isSameMap(m)){
            return true;
        }
    }
    return false;
}

// Class Map
public boolean isSameMap(Map map){
    for (int i = 0; i < getRow(); i++){
        for(int j = 0; j < getCol();
j++){
            if (m[i][j] !=

```

```

map.getIntFromMap(i, j)){
    return false;
}
}
}
return true;
}
}

```

Based on several execution on the program, the output of the program can be displayed such as below:

Table 4.1 Program Output

Number of carbon atoms (n)	Number of Isomers found	Execution Time (ms)	Total Iteration
1	1	0	1
2	1	0	30
3	2	0	780
4	8	66	2441405
5	17	3021	305175780
6	-	> 20 minutes	-

IV. PROGRAM IMPLEMENTATION

Since the program is developed using Java programming language, Java compiler program is needed. The main class will be contained inside the Isomer.java file.

A. Data

Before the isomer search algorithm started, the program will ask the user to input the number of carbon atoms of the alkane compound. The algorithm will start soon after the user finish inputting the number. The output of the process is the amount of isomers, the execution time, and iteration count.

```

Insert the amount of carbon atoms in the Alkane Compound: 2
=====
The Alkane Compound is C2H6
=====
[ 3, 3 ]
[ , ]

=====
1 Isomers found
=====
Execution Time: 0ms
Iteration Count: 30

```

Image 4.1 Program Execution for the n = 2
(Source: Author's archive)

```

Insert the amount of carbon atoms in the Alkane Compound: 3
=====
The Alkane Compound is C3H8
=====
[ 3, 2, 3 ]
[ , , ]

[ 3, 2, ]
[ , 3, ]

=====
2 Isomers found
=====
Execution Time: 0ms
Iteration Count: 780

```

Image 4.2 Program Execution for the n = 3
(Source: Author's archive)

B. Analysis

Based on the data displayed on the previous point, it is implied that the total iteration and the execution time increases significantly as the number of carbon atoms increases. With the n = 5 case, the program needs 3 seconds to complete the calculation while the numbers of total iteration reach 30 million iterations. With 1 different in numbers, the execution time and total iterations might rise greatly.

Therefore, based on the data on Table 4.1 it can be concluded that the Brute Force Algorithm is not suitable to solve this problem. The maximum amount that this algorithm can solve within a considerable amount of time is the 5 amount of carbon atoms inside an alkane compound, or in other words C₅H₁₂.

Furthermore, the program is still lacking because there is still another checking function that needs to be implemented, which is the Nomenclature Checking Function. The isomers solutions displayed by the program still did not considerate the nomenclature of the hydrocarbon compound.

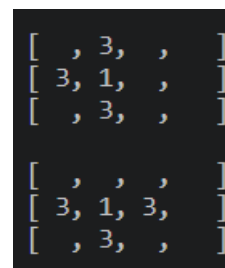


Image 4.3 Isomer Solution of 2-Methylepropane
(Source: Author's archive)

For instance in Image 4.3, we can see that those solution should be counted as one isomer, not two different isomers. Both of the structural compound have the name of 2-Methylpropane. However, just because of the rotation position, the program does not recognize both of the compound as two different compounds.

V. CONCLUSION

Based on the data and analysis on the previous chapter, it can be concluded that the Brute Force Algorithm is not suitable to solve the problem in finding the isomers of an alkane

compounds. The amount of iteration and the execution time increase significantly even with only a difference of one number of carbon atoms. Therefore, in order to solve this specific problem, a more intuitive and intelligent approach might be needed.

VIDEO LINK AT YOUTUBE AND GITHUB

The explanation of the program can be accessed through the link below:

<https://youtu.be/e-eCXnyH03w>

The program data can also be accessed on:

<https://github.com/Gulilil/IsomerFinderBF>

ACKNOWLEDGMENT

First thing first, I would like to sincerely express my gratefulness to the Lord that only because of His grace, this paper could be completed well and submitted within the deadline. Then, I would like to express my gratitude to:

1. All Algorithm Strategies lecturers, especially Ms. Nur Ulfa Maulidevi, for all the lectures that have been given to me on this specific semester,
2. All my closest friends, who have been my support systems in order to go through all the obstacles of this semester, including in writing this paper, and
3. The rightful owners of all the used references of this paper, for every piece of knowledge and information that has been passed upon me.

REFERENCES

- [1] LibreTextsChemistry, "Organic Chemistry." https://chem.libretexts.org/Bookshelves/Introductory_Chemistry/Introductory_Chemistry_%28CK-12%29/25%3A_Organic_Chemistry/25.01%3A_Organic_Chemistry (accessed May 21, 2023).
- [2] "Difference Between Alicyclic and Aromatic Compounds," Jan. 24, 2018. <https://www.differencebetween.com/difference-between-alicyclic-and-vs-aromatic-compounds/#:~:text=Key%20Difference%20E%20%80%93%20Alicyclic%20vs%20Aromatic%20Compounds&text=Aliphatic%20compounds%20have%20single%20or,resulting%20in%20a%20cyclic%20structure.> (accessed May 22, 2023).

- [3] VCEChemistry, "Alkanes, Alkenes, and Alkynes." <https://chemistryvce.weebly.com/alkanes-alkenes-and-alkynes.html> (accessed May 21, 2023).
- [4] LibreTextsChemistry, "Alkanes." [https://chem.libretexts.org/Bookshelves/Organic_Chemistry/Supplemental_Modules_\(Organic_Chemistry\)/Hydrocarbons/Alkanes](https://chem.libretexts.org/Bookshelves/Organic_Chemistry/Supplemental_Modules_(Organic_Chemistry)/Hydrocarbons/Alkanes) (accessed May 22, 2021).
- [5] BYJU'S, "Isomerism." <https://byjus.com/chemistry/isomerism/> (accessed May 22, 2023).
- [6] freeCodeCamp, "Brute Force Algorithms Explained," Jan. 06, 2020. <https://www.freecodecamp.org/news/brute-force-algorithms-explained/#:~:text=Brute%20Force%20Algorithms%20are%20exactly,advanced%20techniques%20to%20improve%20efficiency.> (accessed May 22, 2023).
- [7] R. Munir, "Algoritma Brute Force (Bagian 1)." Accessed: Jan. 23, 2023. [Online]. Available: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-\(2022\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf)

STATEMENT

Hereby, I state that this paper is written on my own, not an adaptation, or translation from other people's paper, and not plagiarism.

Bandung, 12 December 2022



Juan Christopher Santoso - 13521116