String Matching

Vincent Junitio Ungu
18/427597/PA/18557

String Matching?

String Matching tersusun dari kata “String” dan “Matching”.

- String adalah sebuah tipe data yang menyimpan satu karakter atau lebih seperti huruf/ angka.

Contohnya : String tulisan = “Hello World”

- *Matching* adalah sebuah proses mencocokkan sesuatu dengan yang lainnya.

String Matching

Sebuah proses untuk mencocokkan sebuah string dengan string lainnya.

Manfaat

Dapat membantu kita dengan mudahnya mencari ada tidaknya sebuah pattern dalam sebuah text.

Note:

Pattern adalah sebuah string yang digunakan untuk mengecek ada tidaknya kemunculan string ini pada text tersebut.

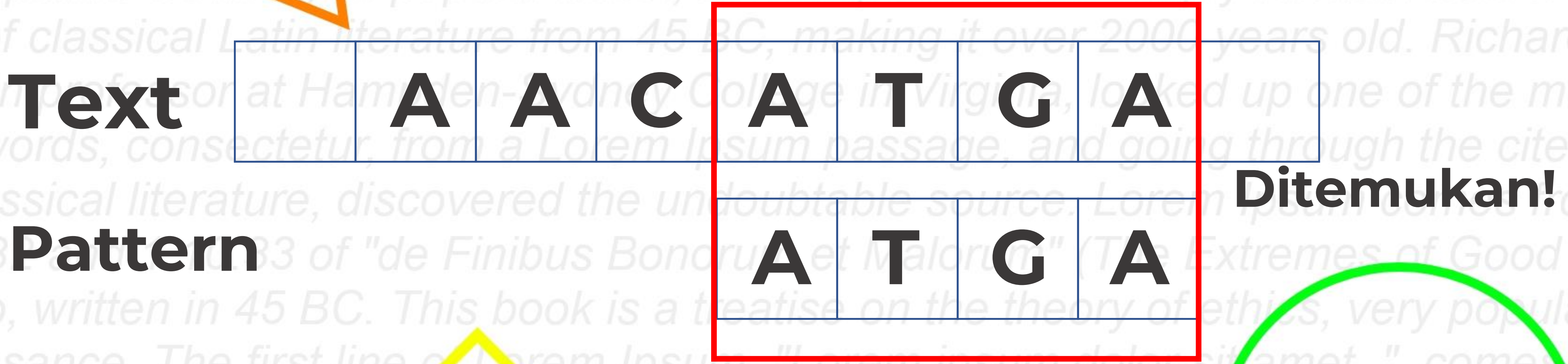
Implementasi

- Mencari urutan DNA
- Mencari kata kunci pada artikel
- Mencari judul buku di perpustakaan

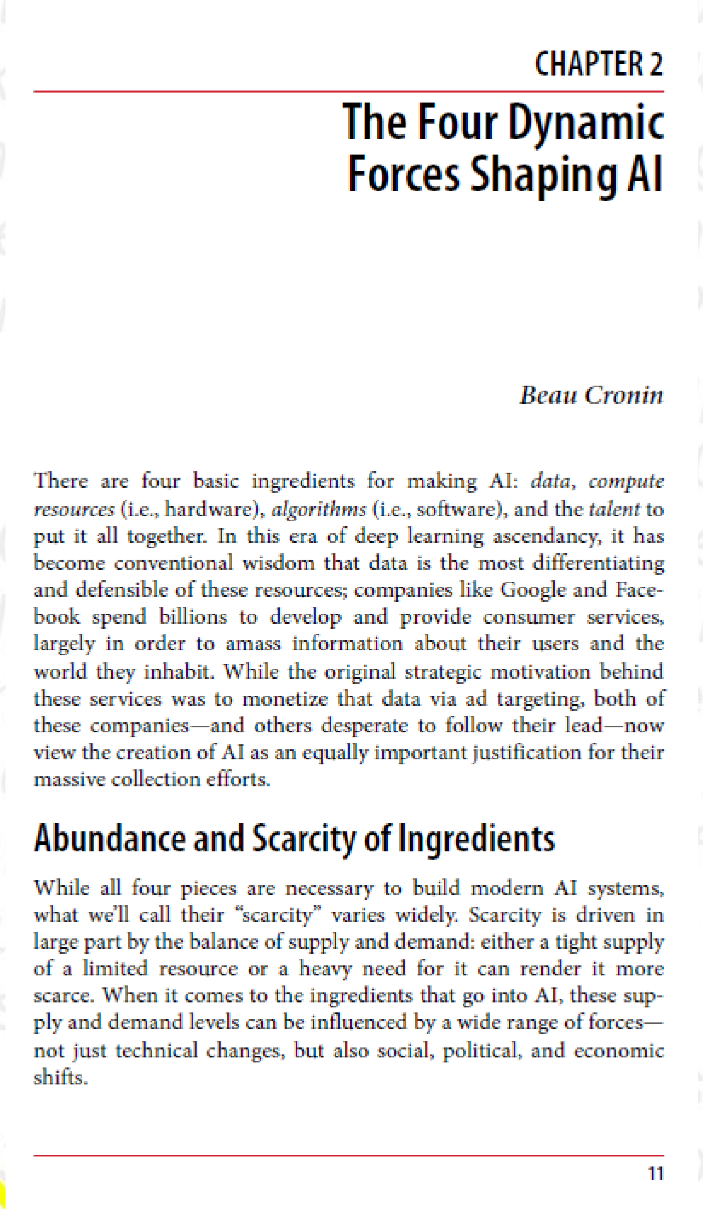
Mencari urutan DNA

Text		A	A	C	A	T	G	A	
Pattern		A	T	G	A				

Mencari urutan DNA



Mencari kata kunci



CHAPTER 2

The Four Dynamic Forces Shaping AI

Beau Cronin

There are four basic ingredients for making AI: *data*, *compute resources* (i.e., hardware), *algorithms* (i.e., software), and the *talent* to put it all together. In this era of deep learning ascendancy, it has become conventional wisdom that data is the most differentiating and defensible of these resources; companies like Google and Facebook spend billions to develop and provide consumer services, largely in order to amass information about their users and the world they inhabit. While the original strategic motivation behind these services was to monetize that data via ad targeting, both of these companies—and others desperate to follow their lead—now view the creation of AI as an equally important justification for their massive collection efforts.

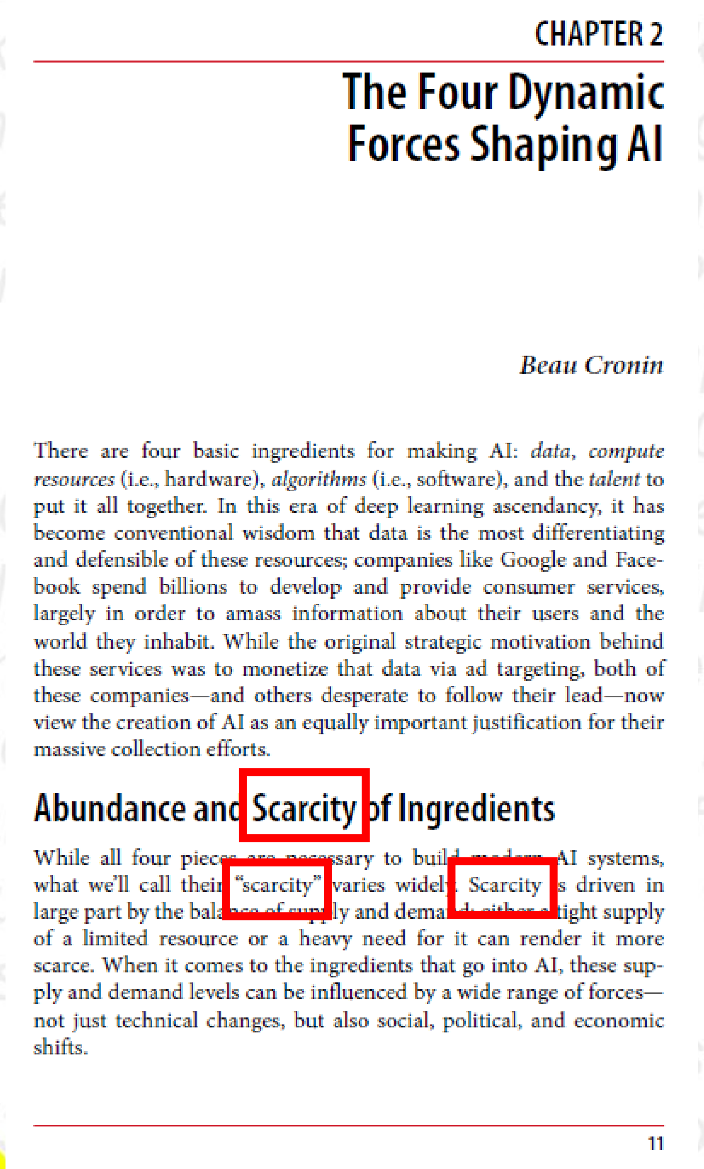
Abundance and Scarcity of Ingredients

While all four pieces are necessary to build modern AI systems, what we'll call their "scarcity" varies widely. Scarcity is driven in large part by the balance of supply and demand: either a tight supply of a limited resource or a heavy need for it can render it more scarce. When it comes to the ingredients that go into AI, these supply and demand levels can be influenced by a wide range of forces—not just technical changes, but also social, political, and economic shifts.

11

Source: Artificial Intelligence Now from O'Reilly Media

Mencari kata kunci



CHAPTER 2

The Four Dynamic Forces Shaping AI

Beau Cronin

There are four basic ingredients for making AI: *data*, *compute resources* (i.e., hardware), *algorithms* (i.e., software), and the *talent* to put it all together. In this era of deep learning ascendancy, it has become conventional wisdom that data is the most differentiating and defensible of these resources; companies like Google and Facebook spend billions to develop and provide consumer services, largely in order to amass information about their users and the world they inhabit. While the original strategic motivation behind these services was to monetize that data via ad targeting, both of these companies—and others desperate to follow their lead—now view the creation of AI as an equally important justification for their massive collection efforts.

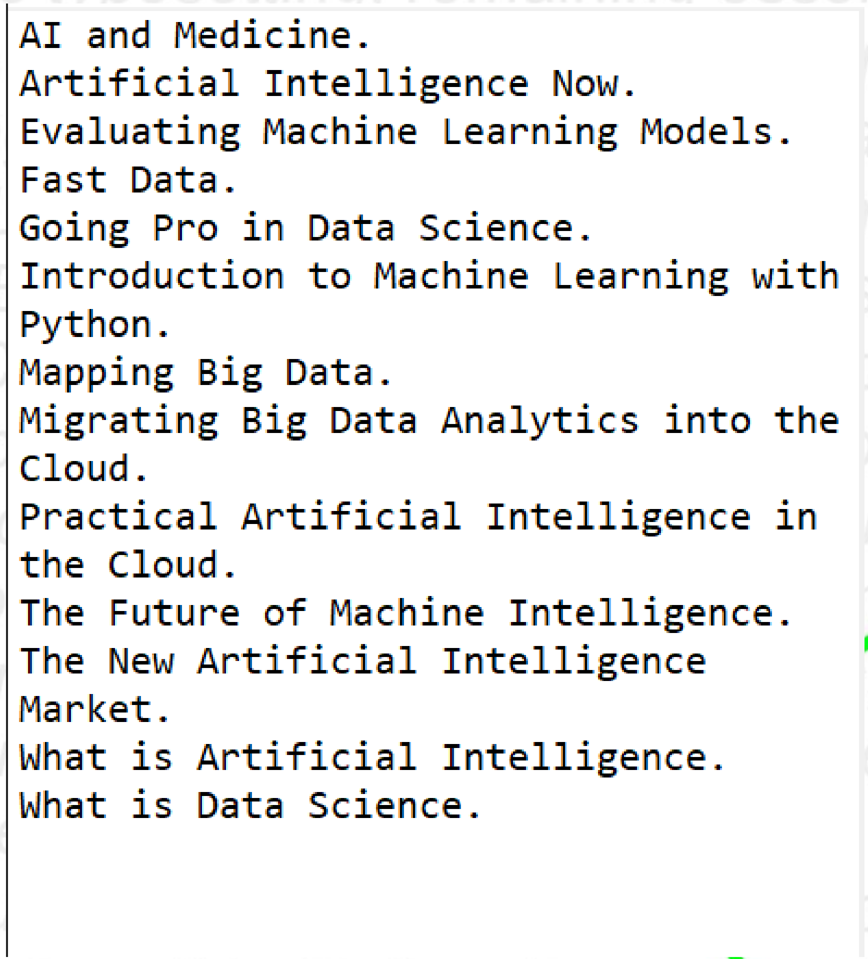
Abundance and Scarcity of Ingredients

While all four pieces are necessary to build modern AI systems, what we'll call their "scarcity" varies widely. Scarcity is driven in large part by the balance of supply and demand: either a tight supply of a limited resource or a heavy need for it can render it more scarce. When it comes to the ingredients that go into AI, these supply and demand levels can be influenced by a wide range of forces—not just technical changes, but also social, political, and economic shifts.

Pattern: scarcity

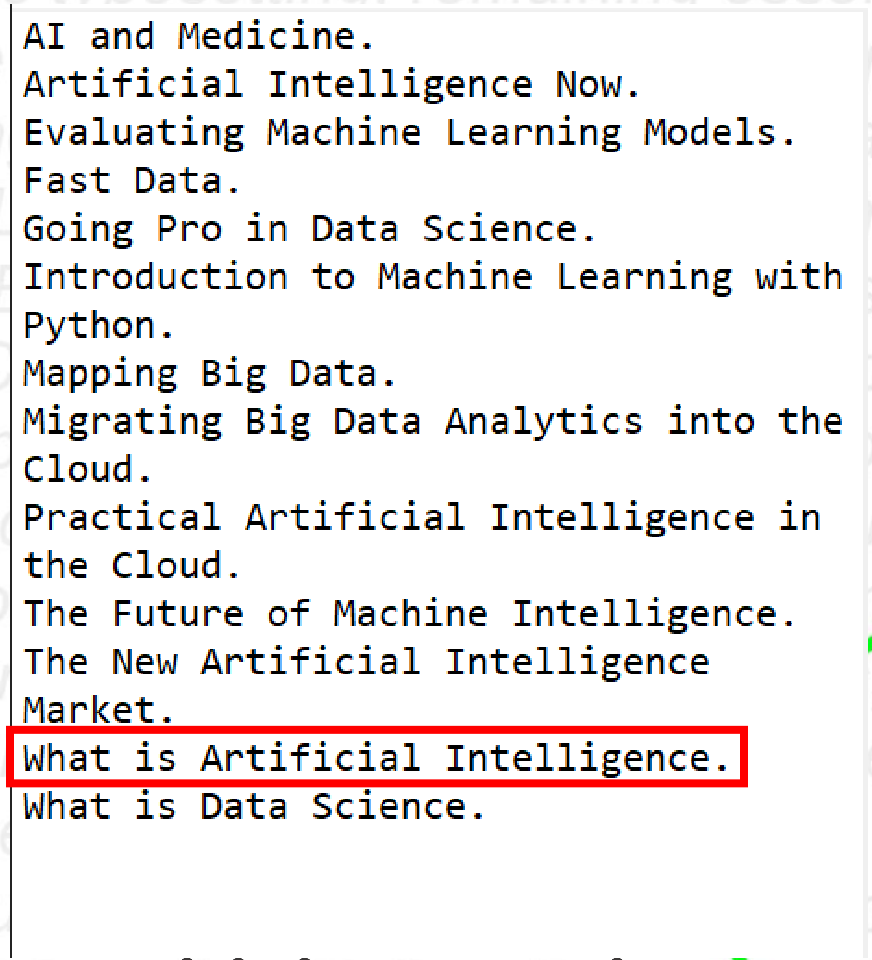
Source: Artificial Intelligence Now from O'Reilly Media

Mencari judul buku



```
AI and Medicine.
Artificial Intelligence Now.
Evaluating Machine Learning Models.
Fast Data.
Going Pro in Data Science.
Introduction to Machine Learning with
Python.
Mapping Big Data.
Migrating Big Data Analytics into the
Cloud.
Practical Artificial Intelligence in
the Cloud.
The Future of Machine Intelligence.
The New Artificial Intelligence
Market.
What is Artificial Intelligence.
What is Data Science.
```

Mencari judul buku



```
AI and Medicine.
Artificial Intelligence Now.
Evaluating Machine Learning Models.
Fast Data.
Going Pro in Data Science.
Introduction to Machine Learning with
Python.
Mapping Big Data.
Migrating Big Data Analytics into the
Cloud.
Practical Artificial Intelligence in
the Cloud.
The Future of Machine Intelligence.
The New Artificial Intelligence
Market.
What is Artificial Intelligence.
What is Data Science.
```

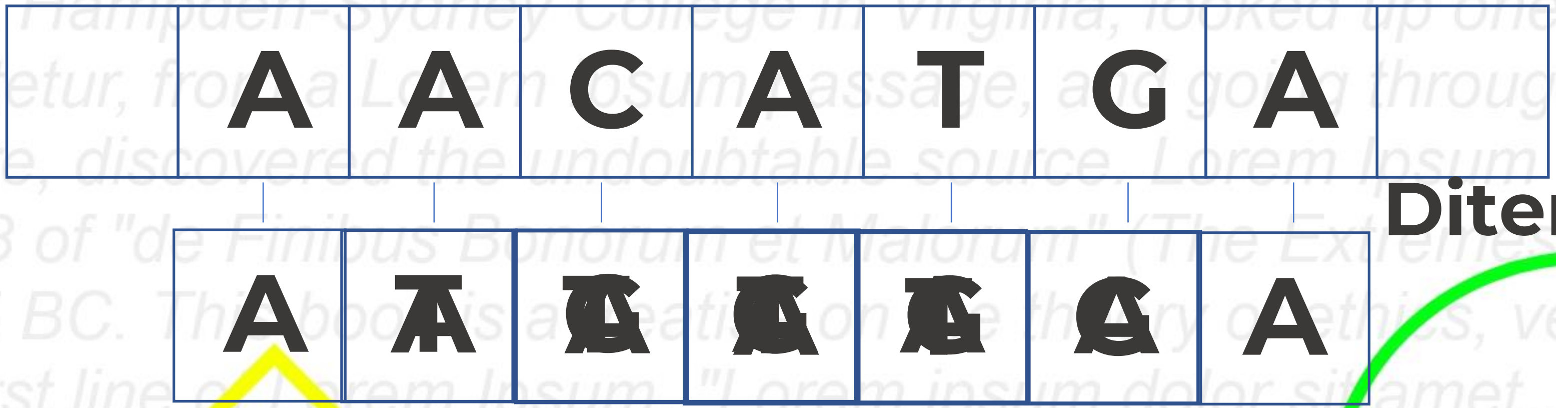
Pattern: What is Artificial Intelligence

Algoritma

- Algoritma Brute Force
- Algoritma Knuth-Morris-Pratt (KMP)
- Algoritma Boyer-Moore
- Algoritma Rabin-Karp

Algoritma Brute Force

- Berdasarkan definisi



Ditemukan!

Algoritma Brute Force

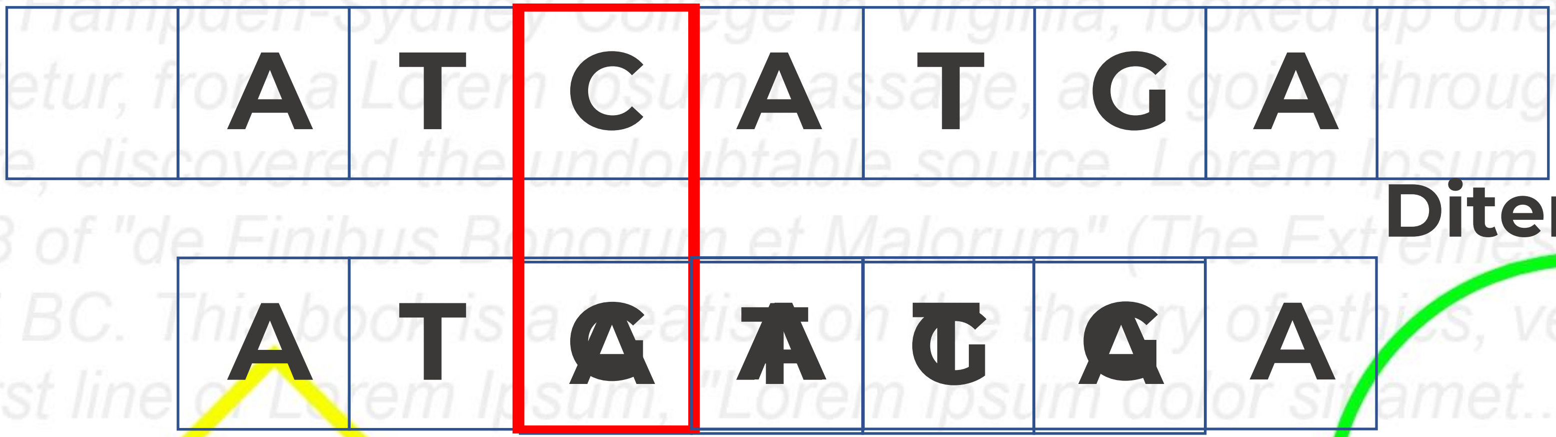
Pseudocode

```
for i=0 to T.length()-P.length(){
    index=i
    for j=0 to P.length(){
        if(T[i]==P[j] and j==P.length()-1){
            i=indexawal;
            return indexawal;
        }else if(T[i]==P[j]){
            i++;
        }else{
            i = indexawal;
            break;
        }
    }
}
```

Kompleksitas Algoritma Brute Force

Algoritma Brute Force akan mengecek satu per satu karakter pattern pada text sehingga worst casenya adalah $\Theta(mn+m-n^2-n)$/ $\Theta(mn)$ dengan m adalah ukuran dari text dan n adalah ukuran dari pattern.

Algoritma Knuth-Morris-Pratt (KMP)



Ditemukan!

Algoritma Knuth-Morris-Pratt (KMP)

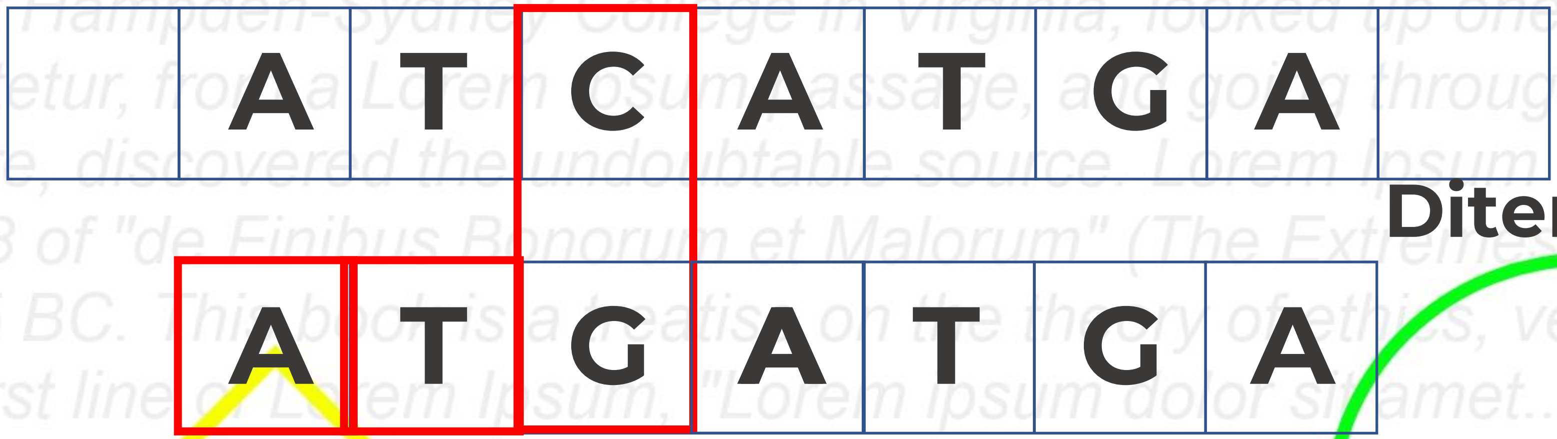
Pseudocode

```
i = 0
j = 0
while(i<T.length()
        if(T[i]==P[j]){
                if(j==P.length()-1){
                        return i-P.length()+1;
                }else{
                        i++; j++;
                }
        }else{
                if(j>0){
                        j = KMPFailure(j-1);
                }
                if(j == null || j == 0) {
                        i++; j=0;
                }
        }
}
```

Kompleksitas Algoritma Knuth-Morris-Pratt (KMP)

Algoritma Knuth-Morris-Pratt (KMP) memiliki kompleksitas waktu ϴ(m) dengan m adalah ukuran dari text.

Algoritma Boyer-Moore



Ditemukan!

Algoritma Boyer-Moore

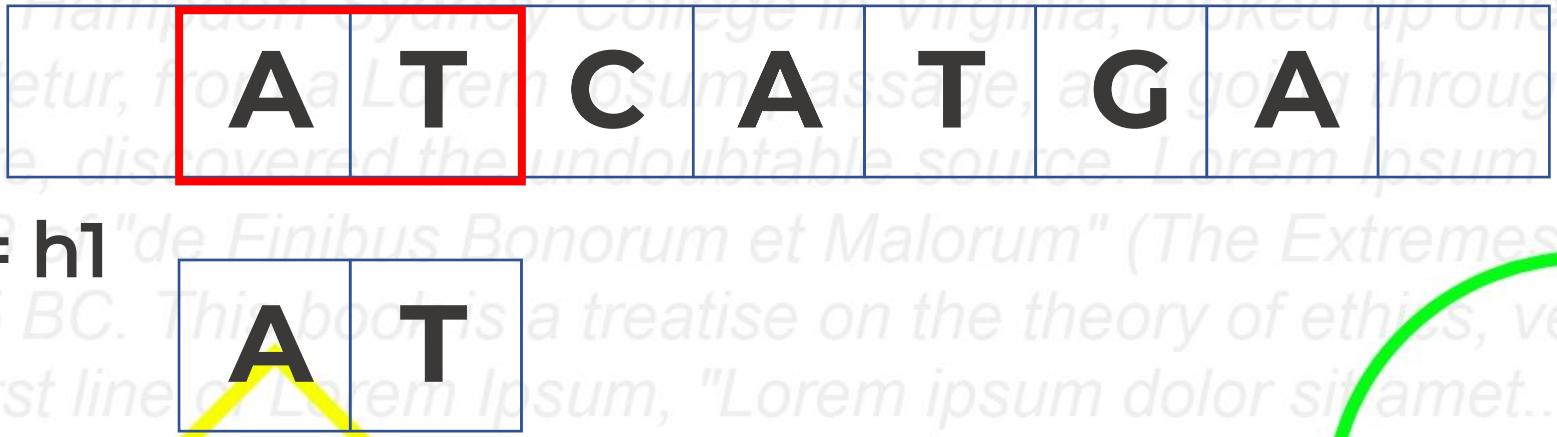
Pseudocode

```
i = P.length()-1
j = P.length()-1
while(i<=T.length()-1){
        if(T[i]==P[j]){
                if(j==0){
                        ++i;
                        return (i-1);
                }else{
                        i--; j--;
                }
        else{
                temp = find T[i] in P[0..i-1];
                if temp != null{
                        shift pattern so that T[i] = P[temp];
                }else{
                        i=i+P.length(); j= P.length()-1
                }
        }
}
```

Kompleksitas Algoritma Boyer-Moore

Algoritma Boyer Moore akan mengecek satu per satu karakter dari index paling belakang pattern pada text sehingga worst casenya adalah ϴ(mn) dengan m adalah ukuran dari text dan n adalah ukuran dari pattern.

Algoritma Rabin-Karp

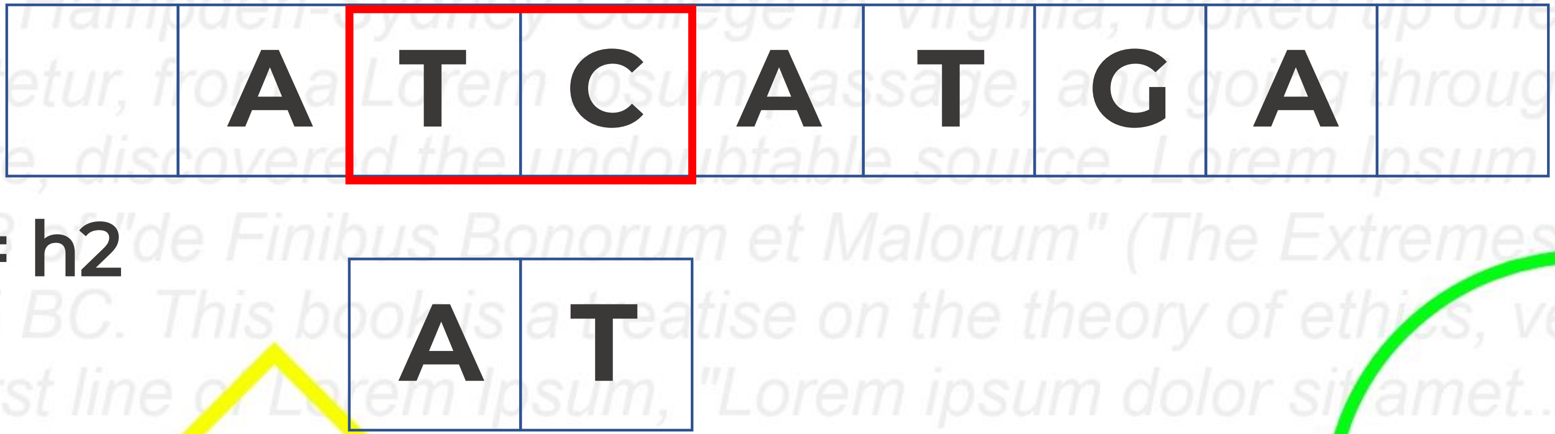


Hash(AT) = h1

Hash(AT) = h1

Ditemukan!

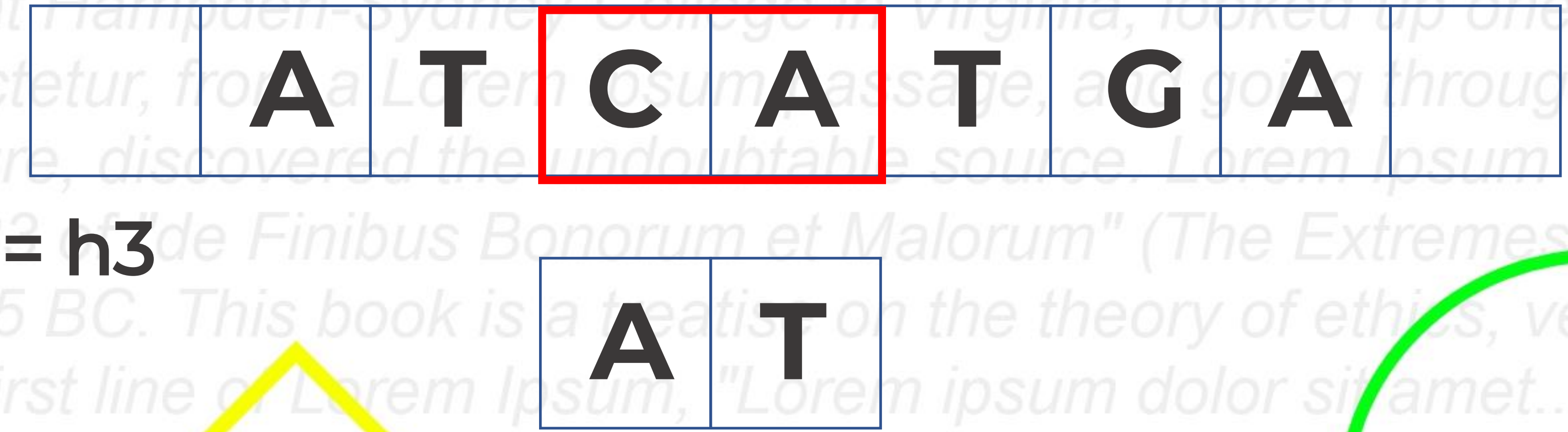
Algoritma Rabin-Karp



Hash(TC) = h2

Hash(AT) = h1

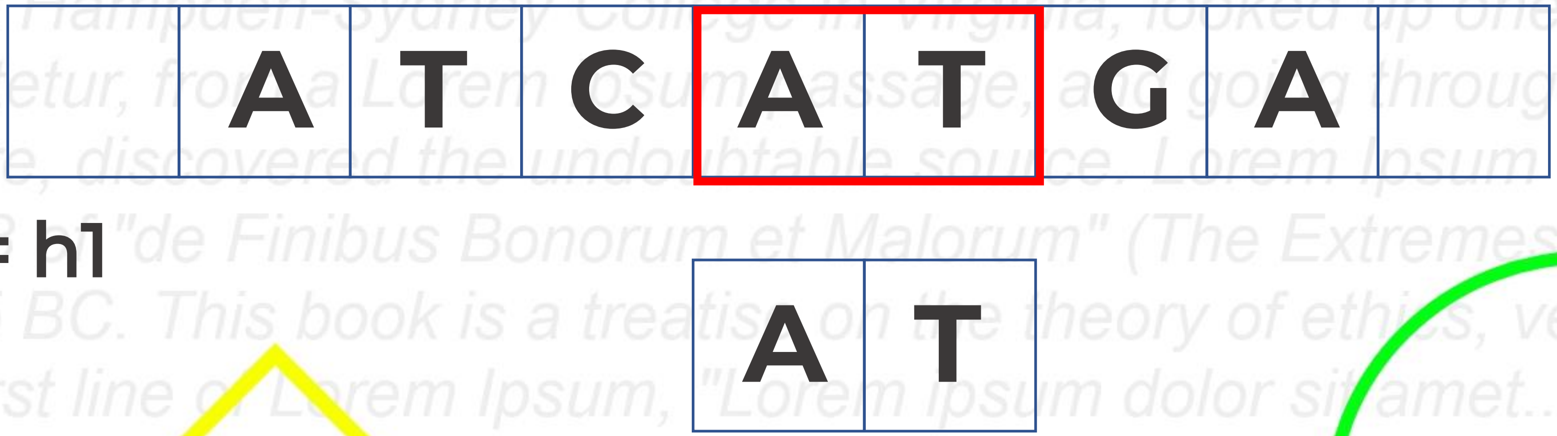
Algoritma Rabin-Karp



Hash(CA) = h3

Hash(AT) = h1

Algoritma Rabin-Karp

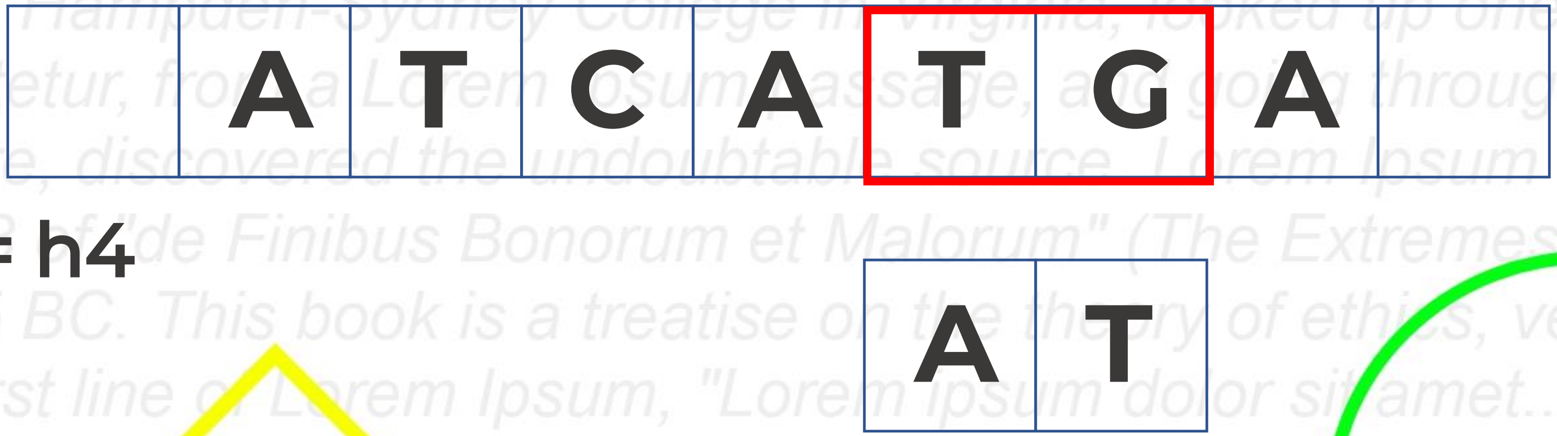


Hash(AT) = h1

Hash(AT) = h1

Ditemukan!

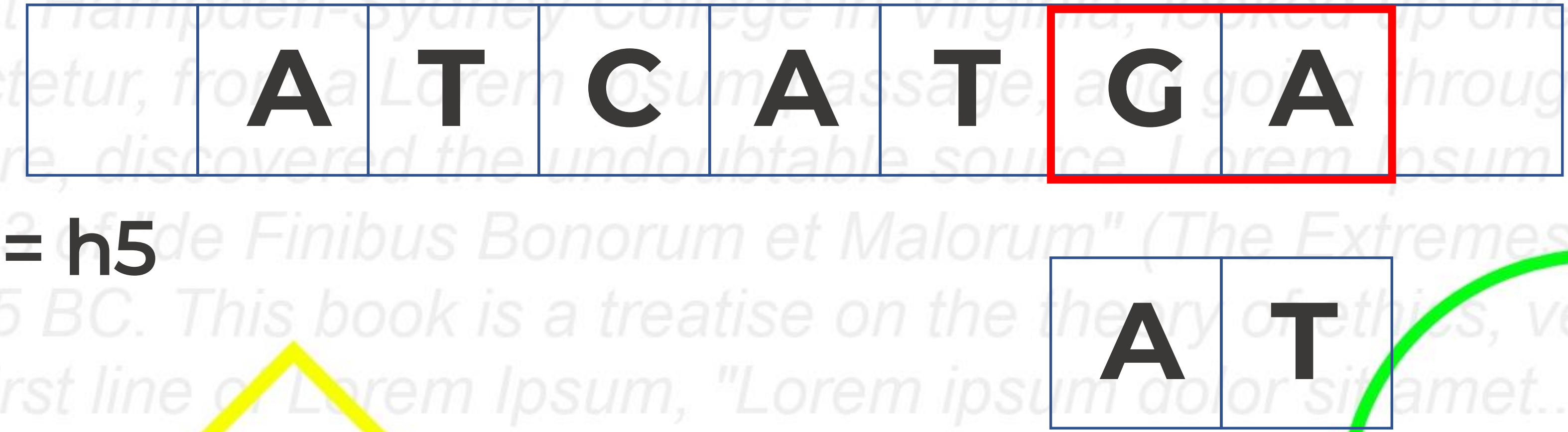
Algoritma Rabin-Karp



Hash(TG) = h4

Hash(AT) = h1

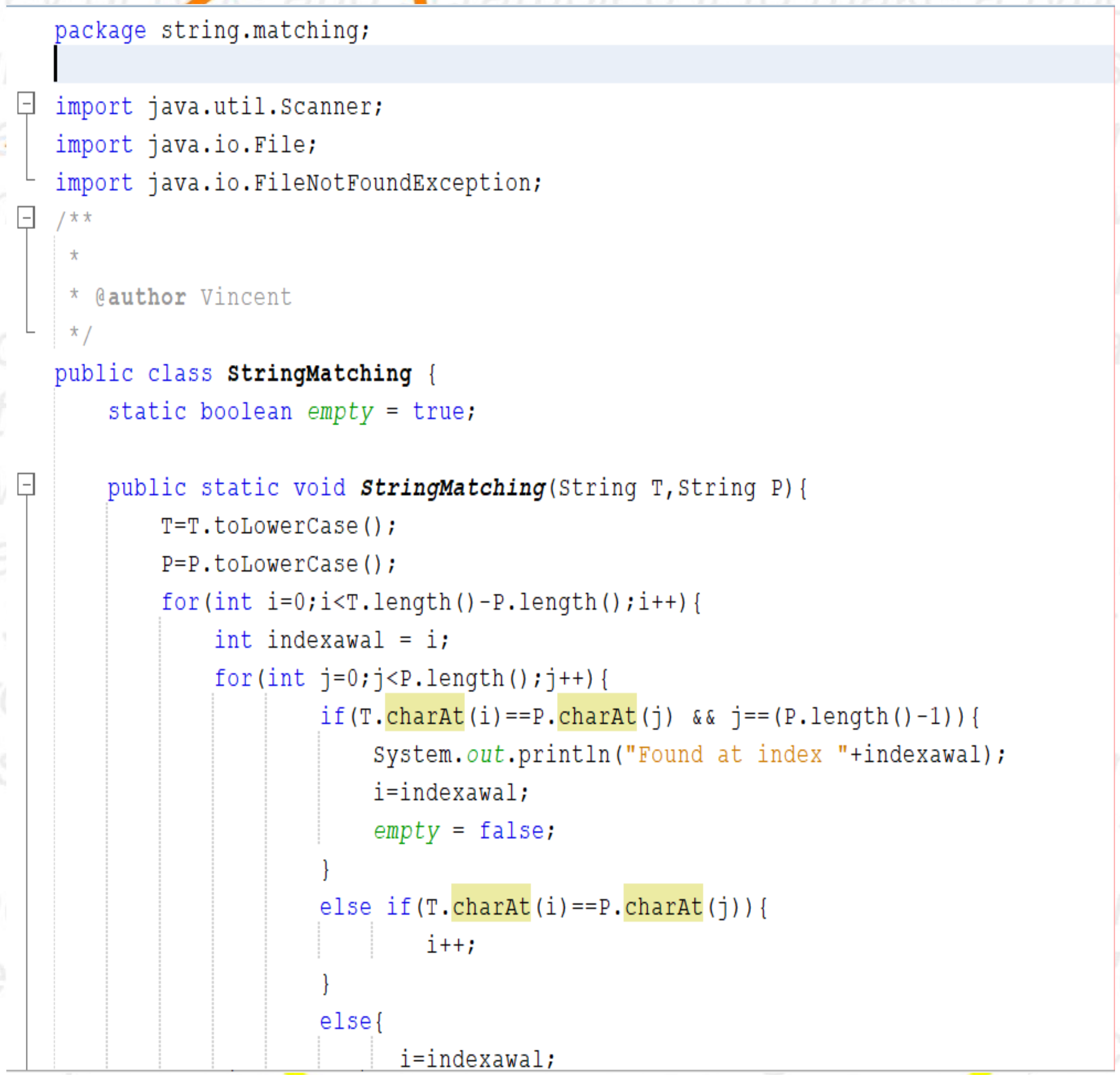
Algoritma Rabin-Karp



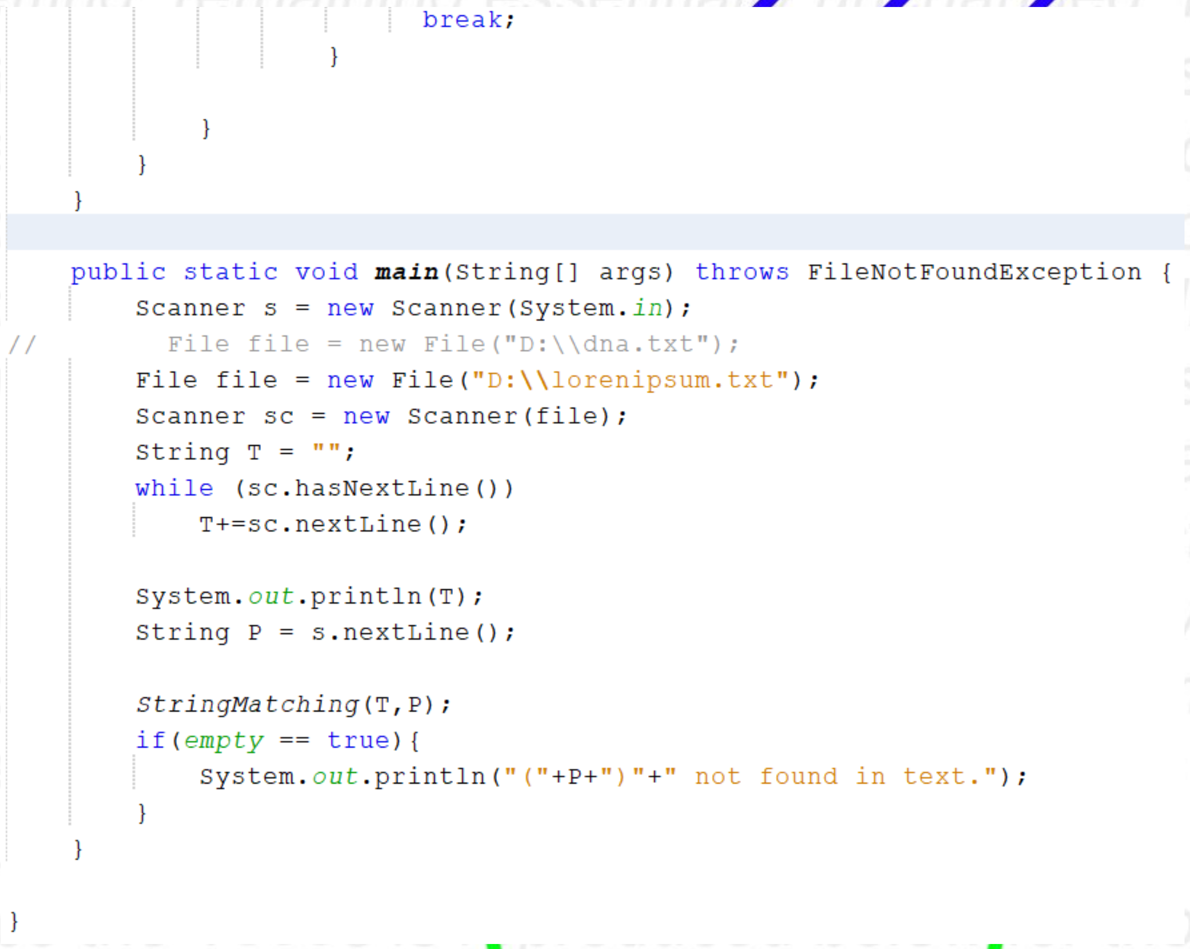
Hash(GA) = h5

Hash(AT) = h1

Program

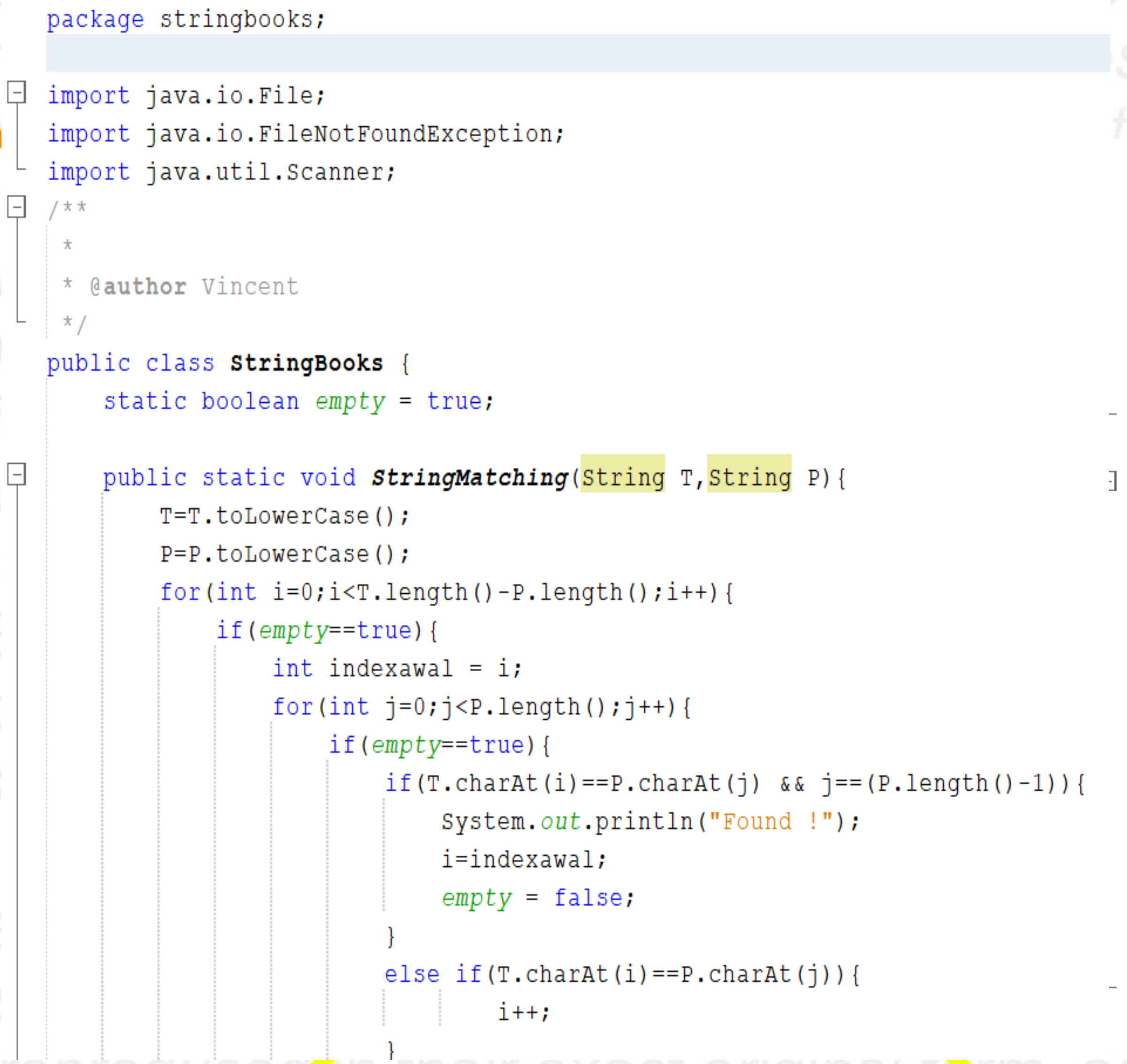


```java
package string.matching;


import java.util.Scanner;
import java.io.File;
import java.io.FileNotFoundException;
/**
 *
 * @author Vincent
 */
public class StringMatching {
    static boolean empty = true;

    public static void StringMatching(String T,String P){
        T=T.toLowerCase();
        P=P.toLowerCase();
        for(int i=0;i<T.length()-P.length();i++){
            int indexawal = i;
            for(int j=0;j<P.length();j++){
                if(T.charAt(i)==P.charAt(j)  && j==(P.length()-1)){
                    System.out.println("Found at index "+indexawal);
                    i=indexawal;
                    empty = false;
                }
                else if(T.charAt(i)==P.charAt(j)){
                    i++;
                }
                else{
                    i=indexawal;
```

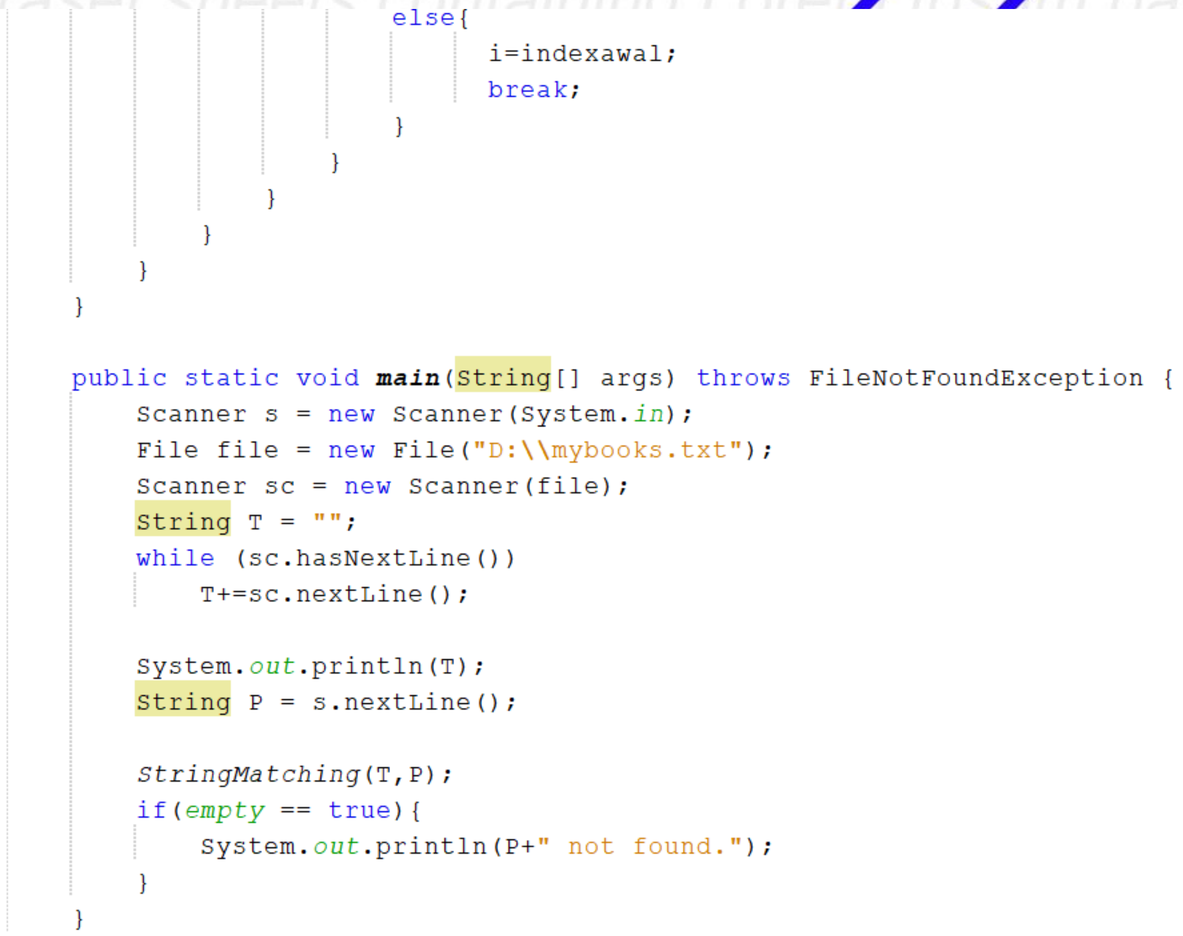


```java
                        break;
                    }

                }
            }
        }

        public static void main(String[] args) throws FileNotFoundException {
            Scanner s = new Scanner(System.in);
//              File file = new File("D:\\dna.txt");
            File file = new File("D:\\lorenipsum.txt");
            Scanner sc = new Scanner(file);
            String T = "";
            while (sc.hasNextLine())
                T+=sc.nextLine();

            System.out.println(T);
            String P = s.nextLine();

            StringMatching(T,P);
            if(empty == true){
                System.out.println("("+P+")"+" not found in text.");
            }
        }

}
```

Program



```java
package stringbooks;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
/**
 *
 * @author Vincent
 */
public class StringBooks {
    static boolean empty = true;

    public static void StringMatching(String T,String P){
        T=T.toLowerCase();
        P=P.toLowerCase();
        for(int i=0;i<T.length()-P.length();i++){
            if(empty==true){
                int indexawal = i;
                for(int j=0;j<P.length();j++){
                    if(empty==true){
                        if(T.charAt(i)==P.charAt(j) && j==(P.length()-1)){
                            System.out.println("Found !");
                            i=indexawal;
                            empty = false;
                        }
                        else if(T.charAt(i)==P.charAt(j)){
                            i++;
                        }
```



```java
                        else{
                            i=indexawal;
                            break;
                        }
                    }
                }
            }
        }
    }

    public static void main(String[] args) throws FileNotFoundException {
        Scanner s = new Scanner(System.in);
        File file = new File("D:\\mybooks.txt");
        Scanner sc = new Scanner(file);
        String T = "";
        while (sc.hasNextLine())
            T+=sc.nextLine();

        System.out.println(T);
        String P = s.nextLine();

        StringMatching(T,P);
        if(empty == true){
            System.out.println(P+" not found.");
        }
    }
```

Thank you for watching

Vincent Junitio Ungu
18/427597/PA/18557