

Nama : Febi rahmadia putri

Nim : 20220021

Matkul : PraktikumDAA

Algoritma Brute Force

Algoritma Sieve of Eratosthenes bekerja dengan cara membangun daftar angka dari 2 hingga batas atas yang ditentukan, dan secara berulang menyaring atau mengeliminasi bilangan-bilangan non-prima dari daftar tersebut.

Berikut adalah contoh kode python dalam algoritma sieve of Eratosthenes

```
def sieve_of_eratosthenes(n):
```

```
    # Inisialisasi array semua bilangan dengan nilai True
```

```
    primes = [True] * (n + 1)
```

```
    primes[0] = primes[1] = False
```

```
    # Saring bilangan non-prima menggunakan algoritma Sieve of Eratosthenes
```

```
    p = 2
```

```
    while p * p <= n:
```

```
        if primes[p]:
```

```
            for i in range(p * p, n + 1, p):
```

```
                primes[i] = False
```

```
        p += 1
```

```
    # Membuat daftar bilangan prima
```

```
    prime_numbers = [num for num, is_prime in enumerate(primes) if is_prime]
```

```
    return prime_numbers
```

```
# Contoh penggunaan
```

```
n = 100
```

```
prime_numbers = sieve_of_eratosthenes(n)
```

```
print("Bilangan prima antara 1 dan", n, "adalah:")  
  
print(prime_numbers)
```

```
1 def sieve_of_eratosthenes(n):  
2     # Inisialisasi array semua bilangan dengan nilai True  
3     primes = [True] * (n + 1)  
4     primes[0] = primes[1] = False  
5  
6     # Saring bilangan non-prima menggunakan algoritma Sieve of Eratosthenes  
7     p = 2  
8     while p * p <= n:  
9         if primes[p]:  
10            for i in range(p * p, n + 1, p):  
11                primes[i] = False  
12            p += 1  
13  
14     # Membuat daftar bilangan prima  
15     prime_numbers = [num for num, is_prime in enumerate(primes) if is_prime]  
16  
17     return prime_numbers  
18  
19  
20 # Contoh penggunaan  
21 n = 100  
22 prime_numbers = sieve_of_eratosthenes(n)  
23 print("Bilangan prima antara 1 dan", n, "adalah:")  
24 print(prime_numbers)  
25
```

Dengan

Hasil

```
Bilangan prima antara 1 dan 100 adalah:  
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

Pada awalnya, array primes diinisialisasi dengan nilai True untuk semua elemen, kecuali indeks 0 dan 1 yang dianggap bukan bilangan prima. Kemudian, algoritma Sieve of Eratosthenes digunakan untuk menyaring bilangan non-prima dengan menandai (mengubah nilai menjadi False) semua kelipatan bilangan-bilangan prima yang ditemukan.

Algoritma Brute Force

Dalam percobaan kedua import itertools ditempatkan di bagian awal program untuk mengimpor modul itertools. Kemudian, fungsi brute_force() akan meminta pengguna untuk memasukkan kata sandi numerik 4 digit dan akan mencoba semua kombinasi mungkin menggunakan itertools.product(). Jika kata sandi ditemukan, akan dicetak jumlah percobaan yang dilakukan. Jika kata sandi tidak ditemukan, pesan "Kata sandi tidak ditemukan" akan dicetak sebelum fungsi brute_force() dipanggil lagi.

```
import itertools
```

```
def brute_force():
```

```
    password = input("Masukkan kata sandi numerik 4 digit: ")
```

```
    chars = "0123456789"
```

```
attempts = 0
```

```
for password_length in range(1, 5):
```

```
    for guess in itertools.product(chars, repeat=password_length):
```

```
        attempts += 1
```

```
        guess = ''.join(guess)
```

```
        if guess == password:
```

```
            print(f"Kata sandi ditemukan setelah {attempts} percobaan.")
```

```
            return
```

```
print("Kata sandi tidak ditemukan.")
```

```
brute_force()
```

```
brute_force()
```

```
1  import itertools
2
3  def brute_force():
4      password = input("Masukkan kata sandi numerik 4 digit: ")
5      chars = "0123456789"
6      attempts = 0
7
8      for password_length in range(1, 5):
9          for guess in itertools.product(chars, repeat=password_length):
10             attempts += 1
11             guess = ''.join(guess)
12             if guess == password:
13                 print(f"Kata sandi ditemukan setelah {attempts} percobaan.")
14                 return
15             print("Kata sandi tidak ditemukan.")
16             brute_force()
17
18  brute_force()
19
```

Dengan hasil yang memunculkan password:

```
Masukkan kata sandi numerik 4 digit: 12345
Kata sandi tidak ditemukan.
Masukkan kata sandi numerik 4 digit: 3456
Kata sandi ditemukan setelah 4567 percobaan.
```

import itertools ditempatkan di bagian awal program untuk mengimpor modul itertools. Kemudian, fungsi brute_force() akan meminta pengguna untuk memasukkan kata sandi numerik 4 digit dan akan mencoba semua kombinasi mungkin menggunakan itertools.product(). Jika kata sandi ditemukan, akan dicetak jumlah percobaan yang dilakukan. Jika kata sandi tidak ditemukan, pesan "Kata sandi tidak ditemukan" akan dicetak sebelum fungsi brute_force() dipanggil lagi.