Analisis Efisiensi Algoritma Iteratif dan Rekursif untuk Penghitungan Bilangan Prima dalam Rentang Tertentu

Universitas Telkom

Disusun Oleh :

Fadhilah Rafif Muflih 103012330395

Ibnu Fajar 103012300122

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY BANDUNG

2024

I. DESKRIPSI STUDI KASUS

Masalah penghitungan bilangan prima dalam rentang tertentu sering ditemukan pada berbagai aplikasi komputasi, seperti kriptografi, pengolahan data, dan teori bilangan. Dalam studi kasus ini, kami membandingkan dua pendekatan algoritmik, yaitu iteratif dan rekursif, untuk menentukan bilangan prima. Analisis ini penting untuk memahami efisiensi kedua metode, terutama ketika diterapkan pada rentang bilangan yang besar.

Selain itu, tujuan utama dari studi ini adalah untuk melihat bagaimana perbedaan struktur algoritma dapat memengaruhi kinerja secara langsung. Dengan memahami kekuatan dan kelemahan dari kedua pendekatan, kami berharap dapat memberikan wawasan yang lebih baik tentang aplikasi praktis dari algoritma ini dalam skenario nyata, termasuk pemrosesan data besar dan optimalisasi komputasi.

II. DESKRIPSI ALGORITMA

Algoritma Iteratif

Pada pendekatan iteratif:

1. Fungsi **isPrimeIterative** memeriksa apakah suatu bilangan adalah bilangan prima. Fungsi ini memanfaatkan pembagian dari angka 2 hingga akar kuadrat bilangan tersebut.
2. Fungsi **primesInRangeIterative** menghasilkan daftar bilangan prima dalam rentang tertentu menggunakan fungsi **isPrimeIterative**. Setiap bilangan dalam rentang diperiksa satu per satu.

Kompleksitas waktu algoritma ini adalah O(n*\sqrt{n}), di mana **n** adalah bilangan maksimum dalam rentang yang diberikan. Pendekatan ini memanfaatkan loop sederhana sehingga mudah diimplementasikan dan cukup stabil untuk ukuran masukan yang besar.

Algoritma Rekursif

Pada pendekatan rekursif:

1. Fungsi **isPrimeRecursive** memeriksa apakah suatu bilangan adalah bilangan prima secara rekursif, mulai dari pembagi terbesar (akar kuadrat bilangan tersebut) hingga 2.
2. Fungsi **primesInRangeRecursive** menghasilkan daftar bilangan prima dalam rentang tertentu dengan cara rekursif, memanfaatkan **isPrimeRecursive**.

Kompleksitas waktu algoritma ini sama dengan iteratif, yaitu O(n*\sqrt{n}), tetapi karena menggunakan rekursi, terdapat overhead tambahan akibat pemanggilan fungsi secara berulang. Struktur rekursif mempermudah penulisan kode, tetapi rentan terhadap batasan memori pada ukuran masukan yang besar.

III. GRAFIK PERBANDINGAN RUNNING TIME

Data Hasil Eksekusi

Rentang Bilangan	Waktu Eksekusi (Iteratif)	Waktu Eksekusi (Rekursif)
1 - 500	0 s	616.3 µs
1 – 1,000	0 s	514 µs
1 – 10,000	2.0482 ms	20.4954 ms
1 – 1,000,000	587.6284	41.0438156 s

Interpretasi Grafik

Berdasarkan hasil eksekusi, pendekatan iteratif secara konsisten lebih cepat dibandingkan pendekatan rekursif. Pada rentang kecil seperti 1-500 dan 1-1,000, waktu eksekusi iteratif hampir tidak terdeteksi karena sangat cepat. Namun, perbedaan signifikan terlihat pada rentang besar seperti 1-10,000 dan 1-1,000,000, di mana waktu eksekusi rekursif meningkat tajam akibat overhead pemanggilan fungsi secara berulang.

Pendekatan iteratif lebih stabil karena hanya memanfaatkan perhitungan langsung tanpa rekursi, sedangkan pendekatan rekursif membutuhkan sumber daya tambahan untuk stack memori, yang menyebabkan waktu eksekusi menjadi lebih lama. Hal ini menegaskan bahwa iteratif adalah pendekatan yang lebih optimal untuk skenario dengan ukuran masukan besar.

IV. ANALISIS PERBANDINGAN KEDUA ALGORITMA

Keuntungan Iteratif:

- Lebih efisien untuk rentang bilangan yang besar.
- Tidak memiliki batasan kedalaman stack seperti rekursif.
- Waktu eksekusi yang lebih cepat pada semua rentang bilangan yang diuji.

Keuntungan Rekursif:

- Kode lebih sederhana dan lebih mudah dibaca untuk rentang kecil.

- Cocok untuk kasus yang membutuhkan pendekatan deklaratif.
- Struktur rekursif membantu memvisualisasikan solusi secara bertingkat.

Kelemahan Iteratif:

- Kode cenderung lebih panjang dan membutuhkan lebih banyak langkah eksplisit.
- Kurang intuitif dibandingkan dengan pendekatan rekursif untuk kasus tertentu.

Kelemahan Rekursif:

- Rentan terhadap stack overflow pada rentang bilangan besar.
- Waktu eksekusi lebih lama karena overhead dari pemanggilan fungsi berulang.
- Tidak optimal untuk aplikasi dengan kebutuhan efisiensi tinggi.

Kesimpulannya, pendekatan iteratif lebih cocok untuk rentang bilangan yang besar karena efisiensinya, sedangkan rekursif lebih cocok untuk rentang kecil yang membutuhkan implementasi sederhana. Hasil ini menunjukkan bahwa pemilihan algoritma harus mempertimbangkan kebutuhan spesifik aplikasi dan ukuran data yang akan diproses.

V. REFERENSI

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", MIT Press, 2009.

Robert Sedgewick, Kevin Wayne, "Algorithms", Addison-Wesley, 2011.

Deskripsi Tugas Besar Mata Kuliah "Analisis Kompleksitas Algoritma" Telkom University.

Artikel online: "Understanding Recursive and Iterative Algorithms" di GeeksforGeeks, https://www.geeksforgeeks.org.

Video Presentasi:

https://drive.google.com/drive/folders/16o4EE8ScVC4PcR2vALeiSf74UiHI828y