Menghitung Siklus Algoritma Dari Collatz Conjecture Dengan Dynamic Programming

Teuku Reza Auliandra Isma

Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
e-mail: reza.auliandra@gmail.com

ABSTRAK

Kemampuan menghitung adalah bagian paling dasar dari komputer. Collatz Conjecture adalah salah satu permasalahan pada bidang matematika yang masih belum bisa dibuktikan kebenarannya. Siklus algoritma dari fungsi Collatz adalah jumlah fungsi yang dilakukan pada bilangan bulat tertentu hingga berhenti pada bilangan 1. Untuk menyelesaikan permasalahan tersebut dengan efisien dapat digunakan *dynamic programming*. Dengan menyimpan hasil dari setiap fungsi Collatz pada tabel, program akan bekerja lebih efisien karena tidak perlu melakukan penghitungan berulang kali untuk hasil yang sama.

Kata kunci: Collatz Conjecture, Dynamic Programming.

1. PENDAHULUAN

Komputer diciptakan pertama kali untuk menyelesaikan berbagai permasalahan di berbagai bidang yang membutuhkan kemampuan untuk menghitung bilangan dalam jumlah besar. Hingga saat ini berbagai komputer dan *software* masih dirancang untuk tujuan tersebut.

Strategi algoritma sebagai salah satu bidang di dalam ilmu komputer memiliki peran yang besar untuk menyelesaikan masalah perhitungan tersebut. Penggunaan strategi algoritma yang tepat dapat meningkatkan kinerja komputer.

Salah satu topik pada strategi algoritma adalah *dynamic programming*. Metode *dynamic programming* dapat diterapkan untuk meningkatkan kinerja komputer dalam menghitung jumlah *cycle* yang dibutuhkan untuk menyelesaikan sebuah fungsi Collatz.

2. DYNAMIC PROGRAMMING

Dynamic Programming adalah metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (step) atau tahapan (stage) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan. Pada *dynamic programming,* rangkaian keputusan yang optimal dibuat dengan menggunakan Prinsip Optimalitas. Prinsip ini berbunyi: *jika solusi total optimal, maka bagian solusi sampai tahap ke-k juga optimal*. Prinsip Optimalitas ini berarti bahwa jika kita bekerja dari tahap k sampai tahap k+1, kita dapat menggunakan hasil optimal dari tahap k tanpa harus kembali ke tahap awal. Jika pada setiap tahap kita menghitung ongkos (cost), maka dapat dirumuskan bahwa:

(cost tahap k+1) = (cost tahap k) + (cost tahap k ke tahap k+1)

Dengan prinsip optimalitas ini, dijamin bahwa pengambilan keputusan pada suatu tahap adalah keputusan yang benar untuk tahap-tahap selanjutnya. Dalam menyelesaikan persoalan dengan program dinamis, orang dapat menggunakan dua pendekatan (*approach*) berbeda: maju (*forward* atau *up-down*) dan mundur (*backward* atau *bottom-up*). Misalkan x_1, x_2, ..., x_n menyatakan peubah (*variable*) keputusan yang harus diambil pada masing-masing tahap 1,2...,n, maka:

1. Program dinamis maju. Program dinamis bergerak mulai dari tahap 1, terus maju ke tahap 2, 3 dan seterusnya sampai tahap n.
2. Program dinamis mundur. Program dinamis bergerak mulai tahap n, terus mundur ke tahap n-1, n-2 dan seterusnya sampai tahap 1.

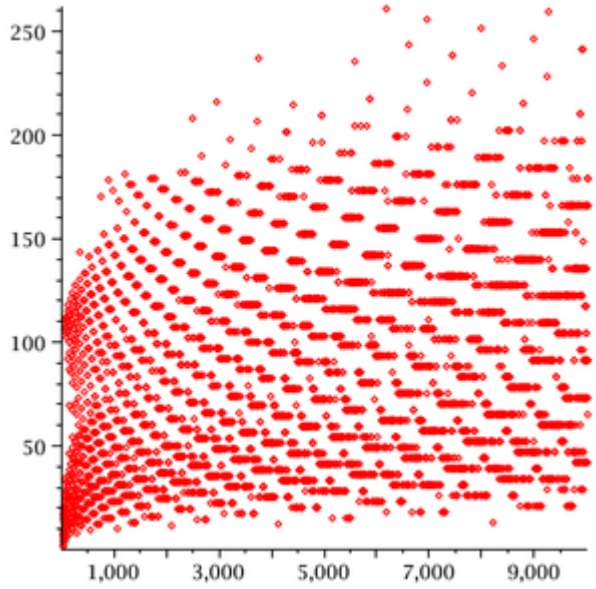
Penyelesaian secara maju atau mundur keduanya ekivalen dan menghasilkan solusi optimum yang sama. Akan tetapi, pengalaman menunjukkan bahwa penyelesaian dengan program dinamis mundur umumnya lebih efisien. Secara umum ada empat langkah yang dilakukan dalam mengembangkan algoritma program dinamis:
1. Karakteristikkan struktur solusi optimal
2. Definisikan secara rekursif nilai solusi optimal
3. Hitung nilai solusi optimal secara maju atau mundur
4. Konstruksi solusi optimal
Perlu dicatat bahwa solusi optimal yang dihasilkan oleh program dinamis dapat lebih dari satu buah.

3. COLLATZ CONJECTURE

Collatz conjecture adalah salah satu permasalahan di bidang matematika yang masih belum dapat diselesaikan. Permasalahan ini diajukan oleh Lothar Collatz pada 1937. Permasalahan ini juga dikenal sebagai 3n + 1 conjecture, Ulam conjecture, Kakutani's problem dan Syracuse problem.
Tentukan sebuah bilangan bulat *n* yang lebih dari 0. Jika n genap maka bagi n dengan 2 (n/2). Jika n ganjil maka kalikan n dengan 3 dan tambahkan 1 (3n + 1). Permasalahannya adalah untuk semua bilangan bulat proses ini akan pada bilangan 1.

$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \equiv 0 \pmod 2 \\ 3n+1 & \text{if } n \equiv 1 \pmod 2 \end{cases}$$

Siklus algoritma adalah jumlah fungsi dibutuhkan untuk mencapai bilangan 1. Collatz conjecture menyatakan bahwa setiap bilangan bulat memiliki siklus algoritma yang terhingga. Misalnya n = 6, maka akan didapatkan barisan 6, 3, 10, 5, 16, 8, 4, 2, 1 dengan siklus algoritma berjumlah 9. Collaz conjecture dapat dinyatakan salah jika didapatkan suatu bilangan bulat n dengan siklus algoritma yang tidak terhingga. Meskipun begitu, hingga saat ini belum ditemukan bilangan bulat yang memenuhi syarat tersebut.



Gambar 1. Grafik siklus algoritma untuk bilangan bulat 0 - 9999

4. DYNAMIC PROGRAMMING PADA COLLATZ CONJECTURE

Dynamic programming digunakan untuk meningkatkan kinerja penghitungan pada fungsi Collatz. Caranya adalah dengan menyimpan jumlah siklus algoritma yang dibutuhkan suatu bilangan bulat pada tabel. Dengan begitu penghitungan tidak harus dilakukan hinga akhir dan tetap memberikan hasil yang benar.

5. KESIMPULAN

Strategi algoritma yang digunakan untuk menyelesaikan permasalahan memiliki pengaruh yang besar terhadap kinerja keseluruhan program. Dynamic programming adalah salah satu algoritma yang paling baik jika dapat digunakan karena tetap memberikan hasil yang optimal dalam waktu yang lebih efisien.

REFERENSI

[1] Rinaldi Munir, *Strategi Algoritma*, Institut Teknologi Bandung, 2007.
[2] http://en.wikipedia.org/Collatz_Conjecture. Waktu akses: 27 Desember 2009