6. Pencacahan Lanjut

Relasi Rekurensi

Relasi rekurensi untuk deretan $\{a_n\}$ adalah persamaan yang menyatakan a_n kedalam satu atau lebih suku sebelumnya, yaitu a_0, a_1, …, a_{n-1}, untuk seluruh bilangan bulat n, dengan $n \geq n_0$, dimana n_0 adalah bilangan bulat tak negatif. Suatu deretan disebut sebagai jawaban dari relasi rekurensi jika suku-sukunya memenuhi relasi rekurensi. Dng kata lain, relasi rekurensi mirip dengan deretan yang didefinisikan secara rekursif, tetapi tanpa menyebutkan nilai (kondisi) awalnya. Maka, relasi rekurensi bisa (dan biasanya) memiliki jawaban ganda (*multiple solution*). Jika kondisi awal dan relasi rekurensi disebutkan dua-duanya, maka deretan dapat ditentukan secara unik.

Contoh 6.1: Tinjau relasi rekursi berikut: $a_n = 2a_{n-1} - a_{n-2}$ untuk $n = 2, 3, 4, \ldots$ Apakah deretan $\{a_n\}$ dengan $a_n = 3n$ merupakan solusi dari relasi rekursi tsb ?

Jawab: Untuk $n \geq 2$ kita peroleh $2a_{n-1} - a_{n-2} = 2(3(n-1)) - 3(n-2) = 3n = a_n$. Jadi, $\{a_n\}$ dengan $a_n = 3n$ adalah jawaban dari relasi rekurensi tsb.

Apakah deretan $\{a_n\}$ dengan $a_n = 5$ solusi dari relasi rekurensi tsb ? Untuk $n \geq 2$ kita memperoleh $2a_{n-1} - a_{n-2} = 2 \cdot 5 - 5 = 5 = a_n$. Jadi, $\{a_n\}$ dengan $a_n = 5$ juga jawaban dari relasi rekurensi tsb.

Pemodelan dengan Relasi Rekurensi

Contoh 6.2: Seorang nasabah menyimpan uang sebanyak \$10.000 di bank dengan suku bunga (majemuk) 5% pertahun. Berapa uang yang dimilikinya setelah 30 tahun ?

Jawab: Mis. P_n jumlah uang di tabungan setelah n tahun. Bagaimana cara menghitung P_n jika P_{n-1} diketahui ? Kita bisa menurunkan relasi rekurensi berikut:

$$P_n = P_{n-1} + 0.05P_{n-1} = 1.05P_{n-1}.$$

Kondisi awalnya adalah $P_0 = 10.000$. Sehingga diperoleh:

$P_1 = 1.05P_0$

$P_2 = 1.05P_1 = (1.05)^2P_0$

$P_3 = 1.05P_2 = (1.05)^3P_0$

…

$P_n = 1.05\mathrm{P}_{n-1} = (1.05)^nP_0$

Kita sekarang punya rumus untuk menghitung P_n untuk tahun ke-*n* dan menghindari iterasi. Dengan menggunakan rumus tsb kita bisa memperoleh besarnya tabungan 30 tahun kemudian, P_{30}, jika tabungan awal (kondisi awal)

$P_0 = 10.000$

…

$P_{30} = (1.05)^{30}\cdot 10.000 = 43.219,42$

Setelah 30 tahun, tabungan tersebut menjadi \$43.219,42.

Contoh 6.3 : Andaikan a_n adalah jumlah bit string yang panjangnya *n* bit tanpa dua bit nol berurutan. String yang demikian kita sebut sebagai *valid strings*. Tentukan relasi rekurensinya dan berikan kondisi awal untuk $\{a_n\}$.

Jawab: [Ide] Banyaknya *valid string* akan sama dengan banyaknya *valid string* yang berakhir dengan 0 ditambah dengan banyaknya *valid string* yang berakhir dengan 1.

Kita asumsikan $n \geq 3$, sehingga string tsb mengandung sedikitnya 3 bit. Selanjutnya kita asumsikan juga bahwa banyaknya *valid string* sepanjang (*n*-1) adalah a_{n-1}, dan banyaknya valid string dng panjang (*n*-2) adalah a_{n-2}. Maka, berapakah banyaknya *valid string* sepanjang *n*, jika string berakhiran bit 1? Ada a_{n-1} string yang demikian, yaitu himpunan valid string sepanjang (*n*–1) dengan bit 1 ditambahkan di akhir string tsb.

Catatan: Menambah 1 pada *valid string* ini tidak mengubah status string tsb (masih *valid string*).

Sekarang kita perlu tahu: ada berapa *valid string* sepanjang n, jika string berakhir dengan 0? *Valid string* sepanjang n yang berakhir dengan 0 haruslah memiliki bit 1 pada posisi ke (n–1) (jika tidak demikian, maka akan berakhir dengan 00). Dan berapa banyak *valid string* sepanjang (n–1) yang berakhiran bit 1? Kita tahu bahwa ada a_{n-1} buah string sepanjang n yang berakhiran bit 1. Jadi, ada a_{n-2} buah string sepanjang (n–1) yang berakhiran bit 1.

Jadi ada a_{n-2} buah *valid string* sepanjang n yang berakhir dng bit 0 (semua *valid string* sepanjang (n–2) dengan bit-bit 10 ditambahkan padanya). Seperti telah disebutkan sebelumnya, banyaknya *valid string* adalah banyaknya *valid string* berakhiran 0 ditambah *valid string* berakhiran 1. Kita mendapatkan relasi rekurensi berikut :

$$a_n = a_{n-1} + a_{n-2}$$

Bagaimana kondisi awalnya?

$a_1 = 2$	{1, 0}
$a_2 = 3$	{11, 10, 01, ~~00~~}
$a_3 = a_2 + a_1 = 3 + 2 = 5$	{111, 110, 101, ~~100~~, 011, 010, ~~001~~, ~~000~~}
$a_4 = a_3 + a_2 = 5 + 3 = 8$	
$a_5 = a_4 + a_3 = 8 + 5 = 13$	
…	

Ternyata deretan ini memenuhi relasi rekurensi seperti pada deretan Fibonacci. Karena $a_1 = f_3$ dan $a_2 = f_4$, diperoleh $a_n = f_{n+2}$.

Memecahkan Persamaan Rekurensi

Pada umumnya, kita lebih memilih memiliki rumus eksplisit untuk mengitung a_n, daripada melakukan iterasi. Formula tersebut bisa diperoleh dengan cara sistematis untuk satu jenis relasi rekurensi. Yaitu relasi rekurensi yang menyatakan suatu suku didalam deretan sebagai kombinasi linier dari suku-suku sebelumnya.

Definisi. Relasi rekurensi homogen derajat k dengan koefisien konstan adalah relasi rekursi berbentuk:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + ... + c_k a_{n-k}$$

dimana c_1, c_2, …, c_k bilangan riil, dan $c_k \neq 0$. Deretan yang memenuhi relasi rekursi yang demikian dapat ditentukan secara unik dengan relasi rekursi dan k buah kondisi awal

$$a_0 = C_0,\ a_1 = C_1,\ a_2 = C2,\ \ldots ,\ a_{k-1} = C_{k-1}$$

Contoh:

- Relasi rekursi $P_n = (1.05)P_{n-1}$ adalah relasi rekurensi linier homogen ber-derajat satu.
- Relasi rekursi $f_n = f_{n-1} + f_{n-2}$ adalah relasi rekurensi linier homogen derajat dua.
- Relasi rekursi $a_n = a_{n-5}$ adalah relasi rekursi homogen linier derajat lima.

Pada saat memecahkan relasi rekurensi yang demikian, kita mencari solusi berbentuk $a_n = r^n$, dimana r konstanta.

$a_n = r^n$ jika a jawaban dari relasi rekurensi

$a_n = c_1 a_{n-1} + c_2 a_{n-2} + ... + c_k a_{n-k}$ jika dan hanya jika $r^n = c_1 r^{n-1} + c_2 r^{n-2} + ... + c_k r^{n-k}$

Kita bagi persamaan ini dengan r^{n-k} dan kurangkan kedua sisi dengan sisi kanan:

$$r^k - c_1 r^{k-1} - c_2 r^{k-2} - ... - c_{k-1} r - c_k = 0$$

Persamaan ini disebut sebagai persamaan karakteristik dari relasi rekursi. Jawaban dari persamaan karakteristik ini disebut sebagai akar karakteristik dari relasi rekurensi. Kita tinjau relasi rekursi homogen linier derajat dua.

Teorema. Misalkan c_1 dan c_2 bilangan riil. Jika $r^2 - c_1 r - c_2 = 0$ mempunyai dua akar berbeda r_1 dan r_2, maka deretan $\{a_n\}$ adalah jawaban dari relasi rekursi $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ jika dan hanya jika $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ untuk n = 0, 1, 2, …, dimana α_1 dan α_2 konstanta.
[Bukti dapat dilihat didalam buku referensi.]

Contoh 6.4: Tentukan jawaban dari relasi rekurensi $a_n = a_{n-1} + 2a_{n-2}$ dengan a_0=2 dan a_1=7 ?
Jawaban: Persamaan karakteristik dari relasi rekurensi tersebut adalah $r^2 - r - 2 = 0$. Akarnya adalah r = 2 dan r = -1. Maka, deretan $\{a_n\}$ adalah jawaban dari relasi rekurensi jika dan hanya jika: $a_n = \alpha_1 2^n + \alpha_2 (-1)^n$ untuk konstanta α_1 dan α_2 tertentu. Dengan persamaan $a_n = \alpha_1 2^n + \alpha_2 (-1)^n$ dan kondisi awal a_0=2 dan a_1=7, diperoleh

$a_0 = 2 = \alpha_1 + \alpha_2$

$a_1 = 7 = \alpha_1 \cdot 2 + \alpha_2 \cdot (-1)$

Pemecahan kedua persamaan diatas memberikan α_1 = 3 dan α_2 = -1. Jadi, jawaban relasi rekursi dan kondisi awal adalah deretan $\{a_n\}$ dengan $a_n = 3 \cdot 2^n - (-1)^n$.

Contoh 6.5: Berikan rumus eksplisit bilangan Fibonacci.
Jawab: Bilangan Fibonacci memenui relasi rekurensi $f_n = f_{n-1} + f_{n-2}$ dengan kondisi awal f_0=0 dan f_1 = 1. Persamaan karakteristinya adalah $r^2 - r - 1 = 0$. Akar-akarnya adalah

$$r_1 = \frac{1+\sqrt{5}}{2} \quad , \quad r_2 = \frac{1-\sqrt{5}}{2}$$

Dengan demikian, bilangan Fibonacci diberikan oleh

$$f_n = \alpha_1 \left(\frac{1+\sqrt{5}}{2} \right)^n + \alpha_2 \left(\frac{1-\sqrt{5}}{2} \right)^n$$

untuk suatu konstanta α_1 dan α_2. Kita bisa menentukan nilai konstanta ini sehingga deretan memenuhi persyaratan $f_0 = 0$ dan $f_1 = 1$:

$f_0 = \alpha_1 + \alpha_2 = 0$

$$f_1 = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right) + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right) = 1$$

Solusi unik sistem dua persamaan dengan dan dua variabel ini adalah

$$\alpha_1 = \frac{1}{\sqrt{5}} \quad , \quad \alpha_2 = -\frac{1}{\sqrt{5}}.$$

Akhirnya rumus eksplisit bilangan Fibonacci ditemukan:

$$\boxed{f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n}$$

Tetapi bagaimana jika persamaan karakteristik itu hanya memiliki satu akar ? Bagaimana mencocokkan persamaan tersebut dengan syarat awal a_0 dan a_1?

Teorema. Misalkan c_1 dan c_2 bilangan riil dimana $c_2 \neq 0$. Andaikan $r^2 - c_1 r - c_2 = 0$ hanya mempunyai satu akar r_0. Deretan $\{a_n\}$ adalah jawaban dari relasi rekursi $a_n = c_1 a_{n-1} + c_2 a_{n-1}$ jika dan hanya jika $a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n$, untuk n = 0, 1, 2, …, dimana α_1 dan α_2 adalah konstanta.

*Contoh 6.6***:** Tentukan jawaban dari relasi rekursi $a_n = 6a_{n-1} - 9a_{n-2}$ dengan $a_0 = 1$ dan $a_1 = 6$?

Jawaban: Satu-satunya akar dari $r^2 - 6r + 9 = 0$ adalah $r_0 = 3$.Maka, jawaban dari relasi rekursi ini adalah $a_n = \alpha_1 3^n + \alpha_2 n 3^n$ untuk konstanta α_1 dan α_2. Untuk memenuhi kondisi awal, diperlukan

$$a_0 = 1 = \alpha_1 \qquad \text{dan } a_1 = 6 = \alpha_1 \cdot 3 + \alpha_2 \cdot 3$$

Pemecahan persamaan ini menghasilkan $\alpha_1 = 1$ dan $\alpha_2 = 1$. Akibatnya, solusi keseluruhan adalah $a_n = 3n + n3^n$.

Relasi "Pecah-Belah-dan-Kuasai"

Ada algoritma yang memecahkan masalah dengan cara memecah masalah tsb kedalam beberapa bagian masalah yang lebih kecil sampai diperoleh jawaban yang trivial. Misalnya,algoritma pencarian biner berturut-turut membagi masukan kedalam dua bagian dan kemudian menghilangkan bagian yang tak relevan hingga tinggal yang relevan. Teknik ini disebut sebagai teknik "divide and conquer" (pecah-belah-dan-kuasai). Kita bisa menggunakan relasi rekurensi untuk menganalisa kompleksitas algoritma yang demikian.

Misalkan suatu algoritma membagi masalah (masukan) berukuran n ke a buah sub-masalah, dimana setiap sub-masalah berukuran n/b. Asumsikan bahwa sebanyak $g(n)$ operasi dilaksanakan untuk setiap bagian masalah tsb. Maka, jika $f(n)$ adalah banyaknya operasi yang diperlukan untuk memecahkan masalah tsb, maka f akan memenuhi relasi rekurensi

$$f(n) = af(n/b) + g(n).$$

Ini disebut sebagai relasi rekurensi "divide-and-conquer" (pecah belah dan kuasai).

Contoh 6.7: Algoritma pencarian biner mengurangi pencarian suatu anggota dari deretan berukuran n ke pencarian biner berukuran $n/2$ (untuk n genap). Diperlukan dua kali proses pembandingan untuk melakukan pengurangan ini. Maka, jika $f(n)$ banyaknya pembandingan yang diperlukan untuk mencari suatu anggota dalam deretan berukuran n, maka

$$f(n) = f(n/2) + 2 \text{ jika } n \text{ genap.}$$

Biasanya, kita tidak mencoba memecahkan relasi *devide-and-conquer* tsb, tetapi menggunakannya untuk menurunkan estimasi big-O dari kompleksitas algoritma.

Teorema. Andaikan f fungsi meningkat yang memenuhi relasi rekurensi $f(n) = af(n/b) + cn^d$ untuk $n = b^k$, dimana k adalah bilangan bulat positif, a, c, dan d bilangan dimana $a \geq 1$, dan b bilangan bulat lebih dari 1. Maka $f(n)$ adalah

$$O(n^d), \qquad \text{jika } a < b^d,$$

$O(n^d \log n)$ jika $a = b^d$,

$O(n^{\log_b a})$ jika $a > b^d$

Contoh 6.8: Untuk pencarian biner, $f(n) = f(n/2) + 2$, sehingga $a = 1$, $b = 2$, dan $d = 0$ ($d = 0$ karena di sini, $g(n)$ tidak bergantung pada n). Akibatnya, $a = b^d$, dan, $f(n)$ adalah $O(n^d \log n) = O(\log n)$. Algoritma pencarian biner memiliki kompleksitas waktu yang logaritmis.