

RECURSION

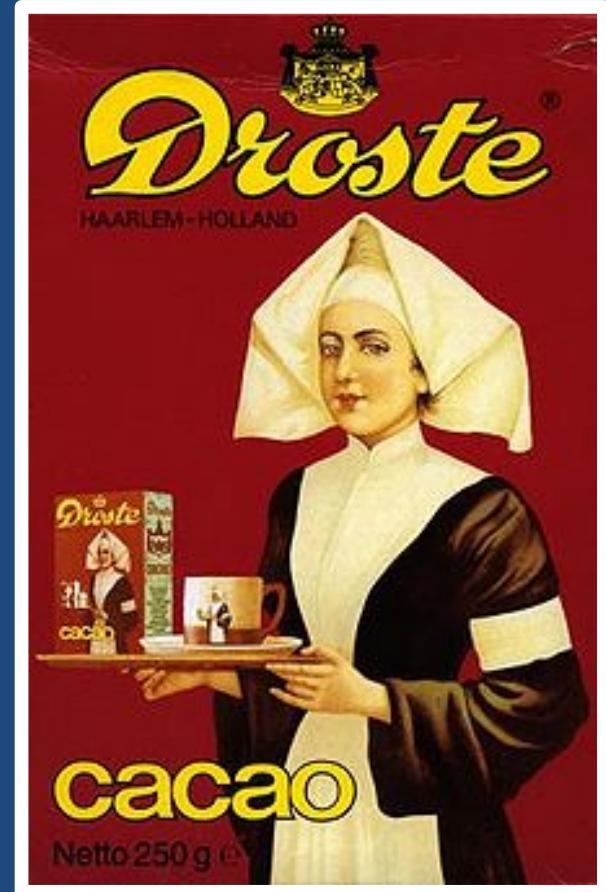
ALGORITMA DAN STRUKTUR DATA
Pertemuan II
Renny P. Kusumawardani, SCJP

Rekursi

- Secara programatik: pemanggilan suatu fungsi oleh dirinya sendiri
- Secara umum: pengulangan suatu hal secara *self-similar*

Contoh Rekursi

- *The Droste Effect*



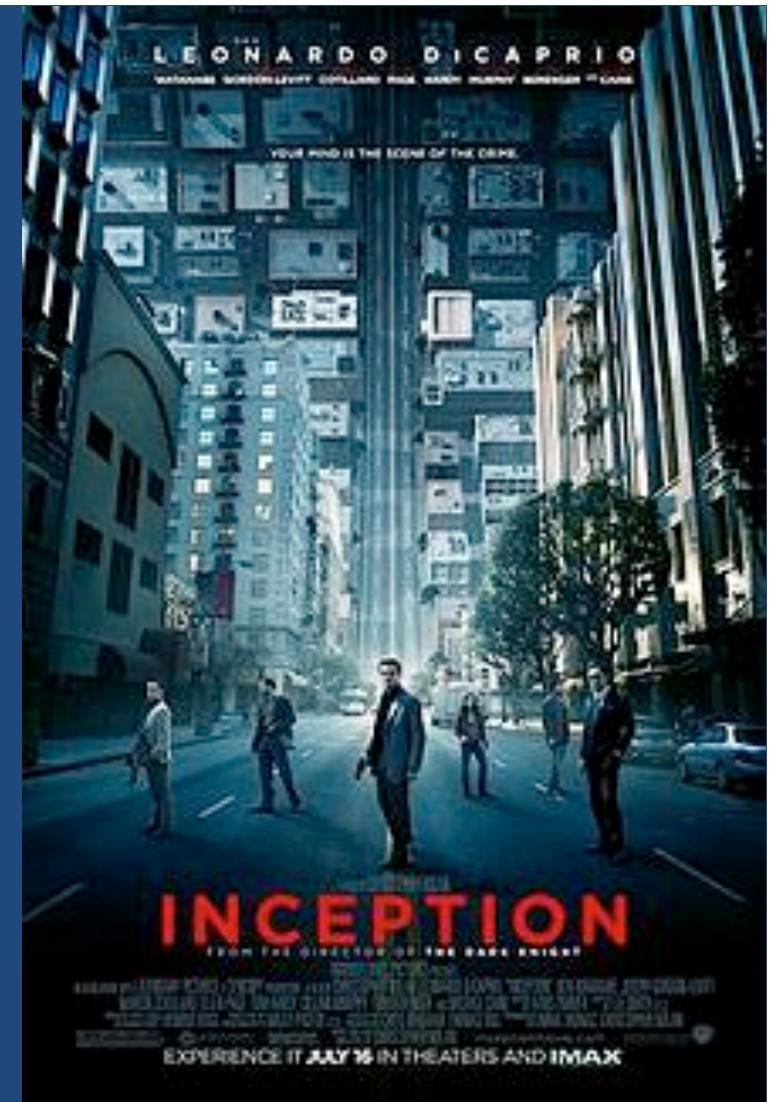
Contoh Rekursi (2)

- Matryoshka Dolls

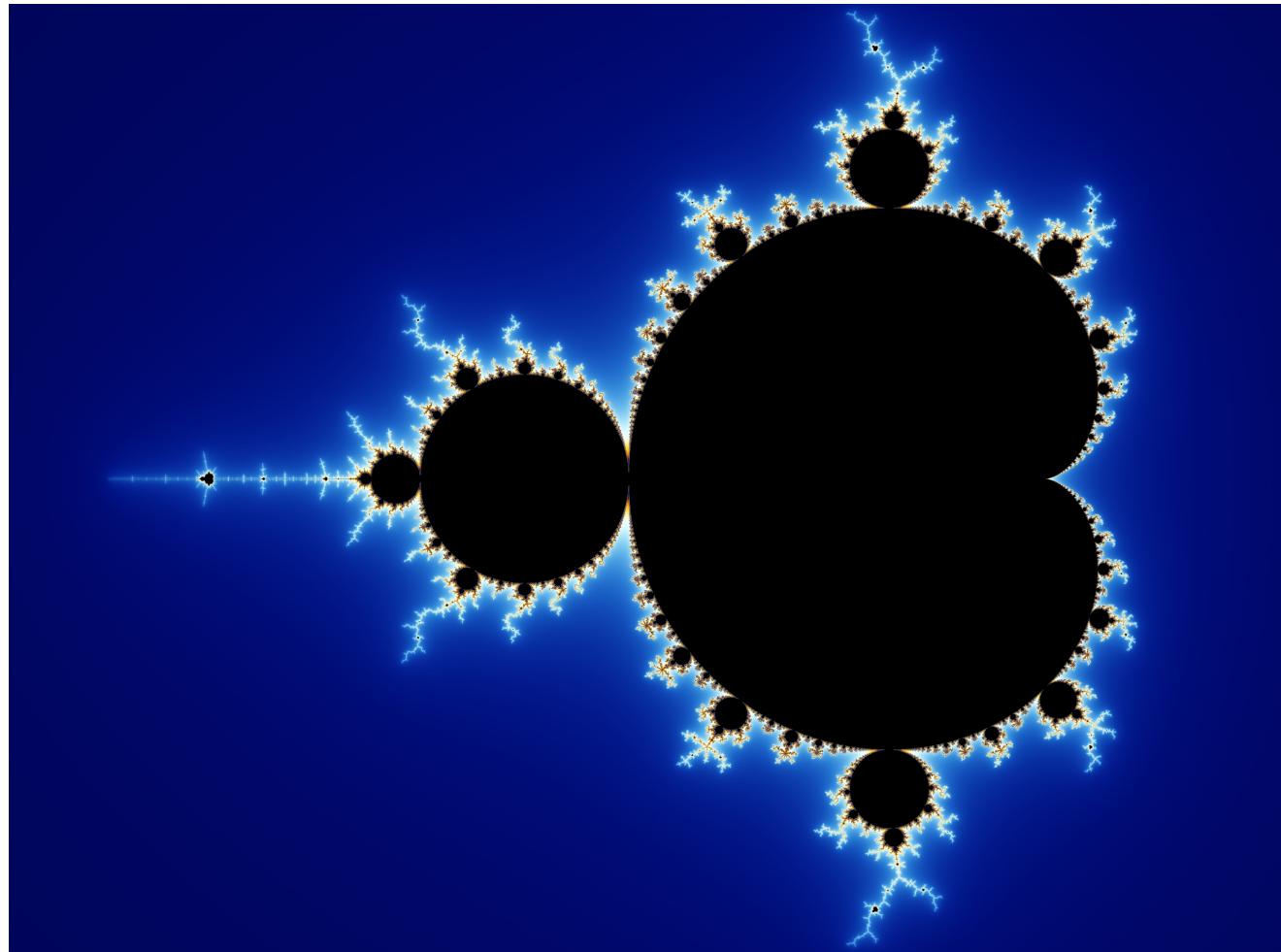


Contoh Rekursi (3)

- **Inception (2010)**
 - Dream within dream







Contoh Rekursi (4)

- The Mandelbrot Set

Recursion: what it means programmatically

- Anda mendefinisikan suatu method yang mengandung pemanggilan terhadap method itu sendiri
- Contoh: faktorial

$$7! = 7 * 6 * 5 * 4 * 3 * 2 * 1$$

$$7! = 7 * 6!$$

$$3! = 3 * 2!$$

$$6! = 6 * 5!$$

$$2! = 2 * 1!$$

$$5! = 5 * 4!$$

$$1! = 1$$

$$4! = 4 * 3!$$

Implementasi Rekursi

$$7! = 7 * 6 * 5 * 4 * 3 * 2 * 1$$

$$7! = 7 * 6!$$

$$6! = 6 * 5!$$

$$5! = 5 * 4!$$

$$4! = 4 * 3!$$

$$3! = 3 * 2!$$

$$2! = 2 * 1!$$

$$1! = 1 \quad \leftarrow \text{disebut pula base case}$$

```
int factorial(int n) {  
  
    if (n == 1) {  
        return 1;  
    } else {  
        return n * factorial(n-1);  
    }  
}
```

Kapan kita menggunakan rekursi?

- Untuk kasus yang dapat dipecah/dinyatakan dalam bentuk yang lebih “sederhana” dari kasus yang sama
- Contoh:
 - **Faktorial:** dapat dinyatakan dalam faktorial yang lebih kecil angkanya
 - **Bilangan Fibonacci:** dapat dinyatakan sebagai penjumlahan bilangan Fibonacci yang lebih kecil

Kapan kita menggunakan rekursi? (2)

- Beberapa kasus rekursi dapat dinyatakan dalam bentuk iterasi, misalnya faktorial
- Rekursi yang dapat dinyatakan dalam iterasi yaitu yang bersifat *tail recursion*
 - suatu rekursi dengan pemanggilan perintah rekursi dilakukan di bagian paling akhir

```
// versi rekursi
int faktorialRekursif(int i) {
    if (i == 1) {
        return i;
    } else {
        return i*faktorialRekursif(i-1);
    }
}

// versi iterasi
int faktorialIteratif(int i) {
    int result = 1;
    for (; i > 0; --i) {
        result *= i;
    }
    return result;
}
```

Anatomi Rekursi

Suatu rekursi terdiri dari komponen-komponen berikut ini:

1. Base case

- Kasus dasar, dengan nilai balikan yang terdefinisi
- Sangat penting, karena dipergunakan untuk penghentian pemanggilan secara rekursif

Contoh:

$\text{faktorial}(1) = 1$, $\text{fibonacci}(0) = 0$, $\text{fibonacci}(1) = 1$

2. Suatu cara untuk memecah permasalahan menjadi lebih sederhana sehingga semakin lama akan semakin mendekati base case

Contoh:

$\text{faktorial}(n) = n * \text{faktorial}(n-1)$ → semakin lama, n akan semakin kecil sehingga angkanya makin mendekati 1

3. Suatu pemanggilan rekursif untuk memanggil hasil dari permasalahan yang lebih sederhana

Contoh:

$\text{faktorial}(n) = n * \text{faktorial}(n-1)$ → terjadi pemanggilan kepada hasil dari kasus yang lebih kecil, yaitu $\text{faktorial}(n-1)$

Contoh

Deret Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

1. Base case:

$$\text{fibonacci}(0) = 0, \text{fibonacci}(1) = 1$$

2. Pemecahan untuk mendekati base case:

$$\text{fibonacci}(n) = \text{fibonacci}(n-1) + \text{fibonacci}(n-2)$$

3. Pemanggilan rekursif

$$\text{fibonacci}(n) = \text{fibonacci}(n-1) + \text{fibonacci}(n-2)$$

Catatan:

pada kasus-kasus yang lebih kompleks, pada definisi fungsi tentunya terdapat statement-statement lain selain pemanggilan fungsi rekursif

Exercise

- Dari <http://codingbat.com/prob/p120015>
- The fibonacci sequence is a famous bit of mathematics, and it happens to have a recursive definition. The first two values in the sequence are 0 and 1 (essentially 2 base cases). Each subsequent value is the sum of the previous two values, so the whole sequence is: 0, 1, 1, 2, 3, 5, 8, 13, 21 and so on. Define a recursive fibonacci(n) method that returns the nth fibonacci number, with n=0 representing the start of the sequence.
- Example outputs:
 - fibonacci(0) → 0
 - fibonacci(1) → 1
 - fibonacci(2) → 1

Latihan (1)

<http://codingbat.com/prob/p183649>

Ada sekumpulan kelinci dan masing-masing memiliki sepasang telinga lebar. Hitunglah jumlah total dari telinga kelinci tersebut secara rekursif, tanpa loop dan tanpa perkalian.

- Misalkan terdapat sejumlah n kelinci, bagaimana memecah persoalan penghitungan ini menjadi persoalan yang lebih kecil?
- Apa base case dari perhitungan rekursi yang dilakukan?
- Tulislah method int telingaKelinci(int n)

telingaKelinci(0) → 0
telingaKelinci(1) → 2
telingaKelinci(2) → 4

Latihan (2)

Sebuah minimarket menata kaleng roti dalam susunan segitiga. Pada lapisan teratas, terdapat satu kaleng roti, pada lapisan kedua terdapat dua kaleng roti, pada lapisan ketiga terdapat tiga kaleng roti, dan seterusnya.

Manajer minimarket tersebut ingin mengetahui, apabila ia menumpuk kaleng roti setinggi n tingkat, berapa banyak kaleng roti yang dibutuhkan. Hitunglah jumlah kaleng tersebut secara rekursif (tanpa loop atau perkalian).

- Jika terdapat n tingkat kaleng roti, bagaimana memecah persoalan penghitungan ini menjadi persoalan yang lebih kecil?
- Apa base case dari perhitungan rekursi yang dilakukan?
- Tulislah method `int jumlahKaleng(int n)`

```
jumlahKaleng(0) → 0  
jumlahKaleng(1) → 1  
jumlahKaleng(2) → 3
```

Latihan (3)

<http://codingbat.com/prob/p184029>

Jika diberikan suatu string, hitunglah secara rekursif (tanpa menggunakan loop) berapa kali string lowercase “hi” muncul pada string tersebut.

- Bagaimana memecah persoalan penghitungan ini menjadi persoalan yang lebih kecil?
- Apa base case dari perhitungan rekursi yang dilakukan?
- Tulislah method int countHi(String str)

countHi ("xxhixx") → 1

countHi ("xhixhix") → 2

countHi ("hi") → 1

countHi ("h") → 0

Latihan 4

- Dengan menggunakan teknik rekursi, jumlahkanlah digit-digit pada suatu bilangan
- Contoh:
 - Jika anda memiliki bilangan 412, hasil yang diharapkan adalah $2 + 1 + 4 = 7$
- Hint:
 - Ingat bahwa $412 / 10 = 41$, $412 \% 10 = 2$
 - Gunakanlah properti pembagian dan modulo pada bilangan bulat ini untuk menuliskan rekursi anda
 - Pertama-tama, cobalah secara manual dengan perhitungan tangan

Reading Assignment

- Subchapters 5.1.2, 5.1.3, 5.1.4
- Subchapter 5.2

Homework

- Buatlah interpretasi dari kode English Ruler
 - Jelaskan makna dari masing-masing fungsi
 - Untuk setiap fungsi, jelaskan makna dari masing-masing perintah