

Analisis Algoritma

Introducing (2)

Hendri Karisma
Program Studi Teknik Informatika
Universitas Komputer Indonesia
2013

Algorithm

*An algorithm is a sequence of unambiguous **instructions** for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time.*

Kenapa algoritma?

Karena kita memiliki masalah atau problem.

Kebutuhan Algoritma

- Combinatorial
- Graph Algorithm
- Network
- Sorting
- Searching
- Learning
- Perceptual Computing
- Recognition
- Information Theory and Digital signal Processing
- Physics
- Bioinformatics
- Big Data
- Etc.

Beberapa point penting pada algoritma

- Kebutuhan nonambiguity untuk setiap langkah pada algoritma tidak dapat ditawar lagi.
- Range input yang dapat diolah oleh algoritma harus ditentukan secara hati-hati
- Algoritma yang sama dapat direpresentasikan dengan cara yang berbeda.
- Selalu ada kemungkinan ada lebih dari satu algoritma untuk menyelesaikan masalah yang sama.
- Algoritma untuk suatu masalah yang sama dapat berdasarkan ide yang berbeda dan algoritma yang berbeda tersebut memiliki kecepatan yang berbeda pula.

Kita tahu algoritma tersebut, namun kapan harus digunakan dan kasus seperti apa yang tepat menggunakan algoritma tersebut?

Bagaimana kita dapat memastikan suatu algoritma memiliki kinerja yang baik dengan kondisi data yang mungkin relatif berbeda?

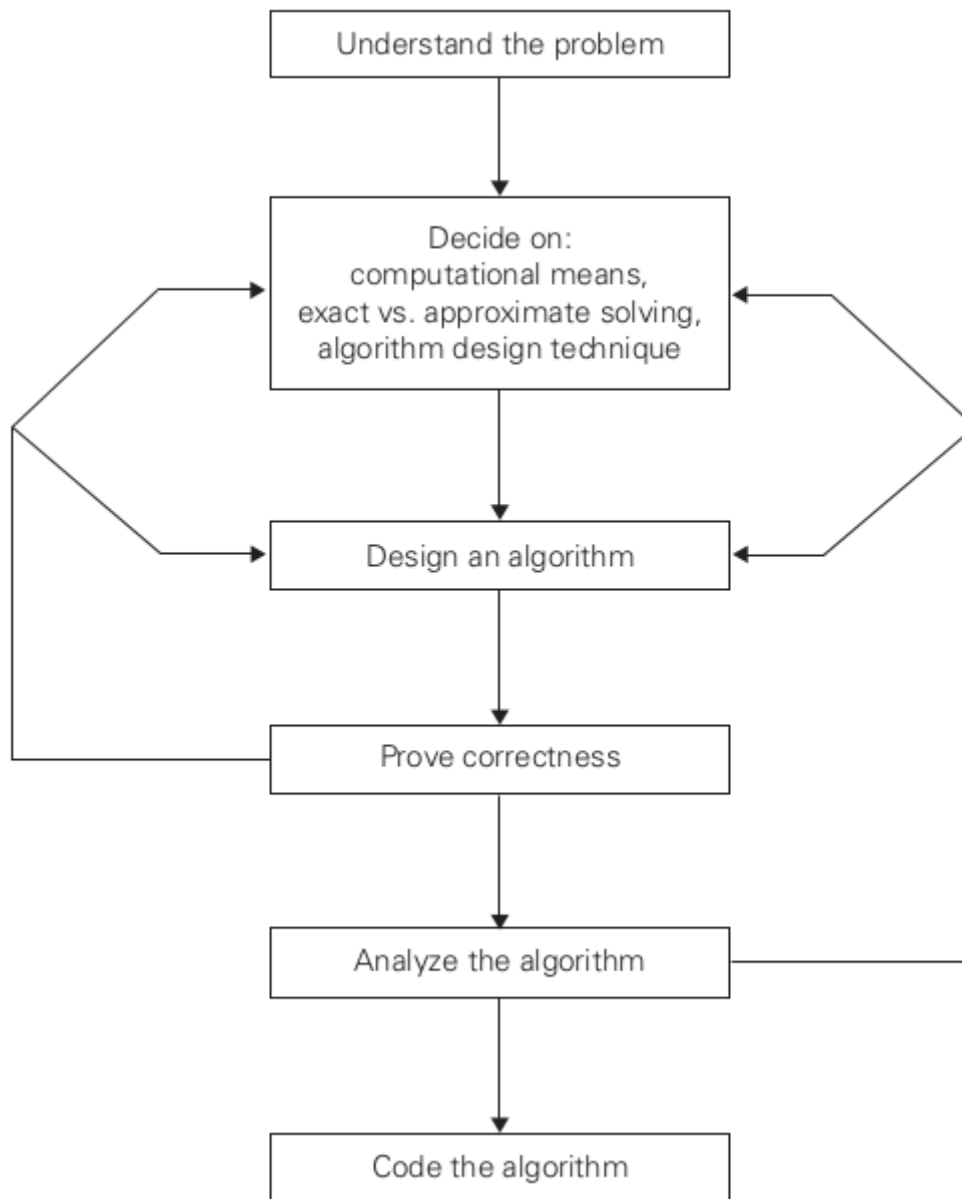
Kenapa Analisis Algoritma?

- Suatu algoritma harus benar (efektif) :
Tujuannya dapat dicapai.
- Suatu algoritma harus dapat berjalan dengan cepat dan lebih ringan (efisien) / mangkus.

...to know how to deal with algorithms:how to construct them, manipulate them, understand them, analyze them... (Donald Knuth)

Fundamentals of Algorithmic Problem Solving

- Understanding the Problem
- Ascertaining the Capabilities of the Computational Device
- Choosing between Exact and Approximate Problem Solving
- Algorithm Design Techniques (strategy)
- Designing an Algorithm and Data Structures
- Methods of Specifying an Algorithm (exp: pseudocode)
- Proving an Algorithm's Correctness
- Analyzing an Algorithm
 - (efficiency, effectivity, simplicity, generality)
- Coding an Algorithm



Analisis Algoritma

- Kompleksitas Algoritma
 - Big O, Big Omega, Big Theta
- Desain dan Analisis Algoritma
 - Brute Force, Greedy, Backtracking, Branch & Bound, Divide & Conquer, Dynamic Programming, etc.
- Pembuktian kebenaran Algoritma (assertion)
 - Computational Logic-based
 - Induksi matematik
 - Hoare Logic

Pembuktian Algoritma

$(x \geq 0) \wedge (y > 0)$
 $= 0;$
 $x := x;$
 while $(r \geq y)$ do
 begin
 $r := r - y;$
 $q := q + 1;$
 end
 $x = q * y + r \wedge (0 \leq r < y)$

$P \wedge B : \{x = q * y + r \wedge (r \geq 0) \wedge (r \geq y)\}$
 $\supset \{x = q * y + r \wedge (r \geq 0) \wedge (r < y)\}$
 $\supset \{x = q * y + r \wedge (0 \leq r < y)\}$
 Q

$(x \geq 0) \wedge (y > 0) \supset \{x = x \wedge (x \geq 0)\}$
 $\supset \{x = 0 * y + x \wedge (x \geq 0)\}$
 $q := 0$
 $\{x = q * y + x \wedge (x \geq 0)\}$
 $r := x$
 $\{x = q * y + r \wedge (r \geq 0)\}$
 $P \wedge B : \{x = q * y + r \wedge (r \geq 0) \wedge (r \geq y)\}$
 $\supset \{x = q * y + r + y - y \wedge (r \geq 0)\}$
 $\supset \{x = q * y + y + r - y \wedge (r \geq 0)\}$
 $\supset \{x = (q + 1) * y + (r - y) \wedge (r - y \geq 0)\}$
 $r := r - y$
 $\{x = (q + 1) * y + r \wedge (r \geq 0)\}$
 $q := q + 1;$
 $\{x = q * y + r \wedge (r \geq 0)\}$

$P \supset (P \wedge B) \supset Q$
 while B do S {Q}

Terbukti
 Terpenuhi

Assignment rule
 $\frac{P \supset R, S \{Q\}}{P \wedge S \{Q\}}$
 $\frac{P \supset R, R \supset Q}{P \supset Q}$