PENGENALAN ANALISIS ALGORITMA

Contoh Masalah

- **Masalah atau persoalan: pertanyaan atau tugas** yang kita cari jawabannya

Contoh :

[Masalah pengurutan] Diberikan senarai (list) S yang terdiri dari n buah data bilangan bulat. Bagaimana mengurutkan n buah data tersebut sehingga terurut secara menaik?

Jawaban dari masalah ini: barisan nilai di dalam senarai yang terurut menaik.

Contoh Masalah

[**Masalah pencarian] Tentukan apakah suatu** bilangan *x terdapat di dalam sebuah senarai S yang* berisi *n buah bilangan bulat!*

Jawaban dari masalah ini: “ya” jika x ditemukan di dalam senarai, atau “tidak” jika *x tidak terdapat* di dalam senarai.

- **Instansiasi masalah: parameter nilai yang** diasosiasikan pada masalah
- Jawaban terhadap instansiasi masalah disebut **solusi**

Contoh: Selesaikan masalah pengurutan untuk

S = [15, 4, 8, 11, 2, 10, 19] n = 7

Solusi: *S = [2, 4, 8, 10, 11, 15, 19].*

Algoritma

- Untuk masalah dengan instansiasi yang besar, solusinya menjadi lebih sulit.
- Perlu sebuah prosedur umum yang berisi langkah penyelesaian masalah

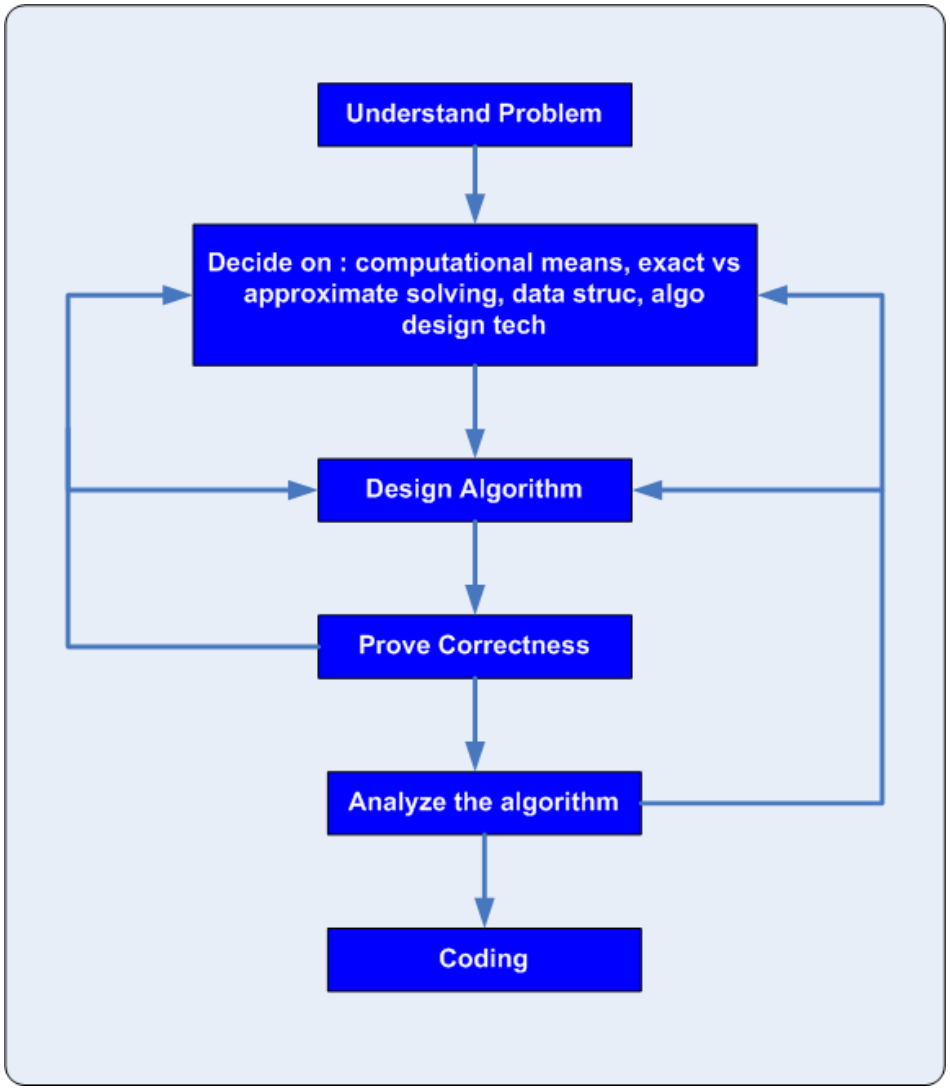
- **Definisi**
  - Algoritma adalah deretan langkah-langkah komputasi yang mentransformasikan data masukan menjadi keluaran[COR92].
  - Algoritma adalah deretan instruksi yang jelas untuk memecahkan masalah, yaitu untuk memperoleh keluaran yang diinginkan dari suatu masukan dalam jumlah waktu yang terbatas. [LEV03].

  Algoritma adalah urutan langkah-langkah untuk memecahkan suatu masalah

Notasi

- Notasi apapun dapat digunakan untuk menuliskan algoritma asalkan mudah dibaca dan dipahami.
- Algoritma dapat ditulis dengan notasi:
  1. Bagan alir (*flow chart)*
  2. Kalimat-kalimat deskriptif
  3. *Pseudo-code (gabungan antara bahasa alami* dengan bahasa pemrograman)

Algorithm Design Process



Penjelasan Design Process

- Input : contoh : menentukan jangkauan masalah
- Kemampuan peralatan komputasi
- Perkiraan :
  - Masalah tidak dapat diselesaikan dengan tepat misalnya akar pangkat dua
  - Algoritma memiliki solusi secara tepat akan ditolak jika memerlukan waktu yang relatif lama
  - Algortima perkiraan adalah bagian dari algoritma eksak yang lebih memuaskan

Penjelasan Design Process

- Algoritma + struktur data = program
- Pendekatan umum untuk memecahkan masalah secara algoritmik.
  - Menentukan algoritma :
  - Menggunakan bahasa alami
  - Menggunakan flowchart
  - Menggunakan pseudocode
  - Menggunakan source code program
  - Menggunakan desain hardware
  - Bentuk lain yang lebih cocok.

Penjelasan Design Process

- Correctness : membuktikan bahwa algoritma menghasilkan hasil yang diinginkan untuk setiap input yang sesuai dalam waktu tertentu
  - Biasanya menggunakan induksi matematis
  - Dapat kita gunakan tracing sederhana
  - incorrectness
  - Approx algotitma → Error < limit

Penjelasan Design Process

- Kualitas algoritma :
  - Correctness
  - Efficiency :
    - Efisiensi waktu
    - Efisiensi ruang
- Simplicity
- Generality : masalah dan rentang inputan
- Pemrograman Algoritma :
  - Resiko : transisis yang tidak benar atau tidak efisien
  - Pembuktian kebenaran program
  - Practical : testing dan debugging

Algorithm Design Techniques

pendekatan umum untuk memecahkan masalah secara algoritmis yang dapat diterapkan pada beraneka ragam masalah guna mencapai tujuan

Klasifikasi strategi algo

1. Strategi solusi langsung (*direct solution strategies)*
   - Algoritma *Brute Force*
   - Algoritma *Greedy*
2. Strategi berbasis pencarian pada ruang status (*state-space base strategies)*
   - Algoritma *backtracking*
   - Algoritma *Branch and Bound*

3. Strategi solusi atas-bawah (*top-down solution strategies)*

   - Algoritma *Divide and Conquer.*

4. Strategi solusi bawah-atas (*bottom-up solution strategies)*

   - *Dynamic Programming.*

Problem Types

- Sorting
- Searching
- String processing
- Graph problem
- Combinatorial problem
- Geometric Problem
- Numerical Problem

Tipe Problem : Sorting

- Problem : menyususn ulang hal-hal yang terdapat pada daftar dengan urutan naik.
- Jika ada records , kita perlu sebuah key
- Terdapat beberapa lusin algoritma sorting
- Dua properti algortima sorting :
  - Stabil : Mempertahankan urutan relatif sembarang dua elemen input yang sama
  - Di tempat : tidak memerlukan memori ekstra kecuali , mungkin beberapa unti memori

Tipe Problem :Searching

- Masalah : menemukan suatu nilai dari sekumpulan nilai yang ada.
- Jangkauan algoritma searching :
  - Pencarian sekuensial hingga binary (sangat efisien, namun terbatas) dan algoritma didasarkan pada representasi kumpulan nilai tersebut sehingga memungkinkan pencarian yang lebih baik.
- Tantangan :
  - Kumpulam data yang sangat besar
  - Update : add, edit, delete

Tipe Problem : Pemrosesan String

- String = urutan karakter alphabet
- Minat Khusus : text strings, binary strings dan lain-lain
- Problem yang khusus : pencocokan string
  - Pencarian suatu kata dalam text

Tipe Problem : Graph Problem

- Algoritma graph dasar : graph traversal, shortest-path, sorting topologik pada graph dengan ujung berarah
- Beberapa masalah sangat sulit diselesaikan dengan cara komputasi hanya beberapa contoh yang dapat diselesaikan dalam waktu yang dapat diterima. Contoh :
  - ❑TSP (Traveling Salesman Problem)
  - ❑GCP(Graph Coloring Problem) → Pewarnaan Graph

Tipe Problem : Combinatorial Problem

- Masalah : Menemukan suatu objek kombinatorik seperti permutasi, kombinasi atau subset yang memenuhi batasan tertentu dan memiliki properti yang diinginkan.
- Problem yang paling sulit :
  - Sejumlah objek kombinatorik tertentu tumbuh dengan cepat seiring peningkatan ukuran masalah.
  - Tidak diketahui algoritma eksak untuk menyelesaikan masalah tersebut.
- Salah satu contohnya TSP dan GCP.

Tipe Problem : Geometric Problem

- Berkaitan dengan objek geometrik : titik, garis. poligon dan lain-lain
- Yunani Kuno : membangun geometrik sederhana contohnya segitiga, lingkaran dan lain-lain
- Masa kini : aplikasi komputer grafik, robot
- Masalah klasik :
  - Problem closest pair : diberikan titik pada suatu bidang, dan temukan pasangan terdekatnya
  - Convex hull : temukan poligon cembung terkecil yang melibatkan smeua titik yang telah ditentukan

Tipe Problem : Numeric Problem

- Berkaitan dengan objek matematis yang memiliki sifat kontinyu: memecahkan persamaan dan sistem persamaan, menghitung integral tak hingga dan lain-lain
- Mayoritas permasalahan diatas dapat dipecahkan dengan perkiraan.
  - Komputer hanya akan merepresentasi angka real dengan kira-kira.
  - Akumulasi kesalahan round-off
- Perubahan fokus komputasi industri : analisis numerik (pada industri dan ilmu pengetahuan) menuju aplikasi bisnis (penyimpanan informasi, transportasi melalui jaringan dang presentasi kepada pengguna)

Analisis Algoritma

- Sebuah algoritma tidak hanya harus benar, tetapi juga harus mangkus (*efficient)*
- Ukuran kemangkusan algoritma: waktu dan ruang memori (*space).*
- Algoritma yang mangkus: algoritma yang meminimumkan kebutuhan waktu dan ruang

Alat ukur efesiensi algoritma

- Alat ukur kemangkusan algoritma:
  1. Kompleksitas waktu, *T(n)*
  2. Kompleksitas ruang, *S(n)*
- *n = ukuran masukan yang diproses oleh algoritma*
- *T(n) : jumlah operasi yang dilakukan untuk* menjalankan sebuah algoritma sebagai fungsi dari ukuran masukan *n.*
- *S(n): ruang memori yang dibutuhkan algoritma* sebagai fungsi dari ukuran masukan *n*