Computational Thinking

(General Introduction)

Video 5.0a

Hon Wai Leong
Department of Computer Science
National University of Singapore

Email, FB: leonghw@comp.nus.edu.sg

Learn CT & Develop ITeMS

Content of this video

- ⇨ □ Components of the CT Segment
- □ Quick Self-Introduction
- □ Quick intro to CS
- □ CT is for Everyone, ITeMS
- □ Problem Solving via “Asking Questions”

Components of the CT Segment

- ❑ (this video) Intro to CT Segment
- ❑ Lectures 5 & 6 (few short videos each)
- ❑ Tutorial (problem solving)
  - ❖ some discussed, some to be handed-in
- ❑ Compulsory Readings (2)
- ❑ Miscellaneous
  - ❖ A few videos on CT related topics

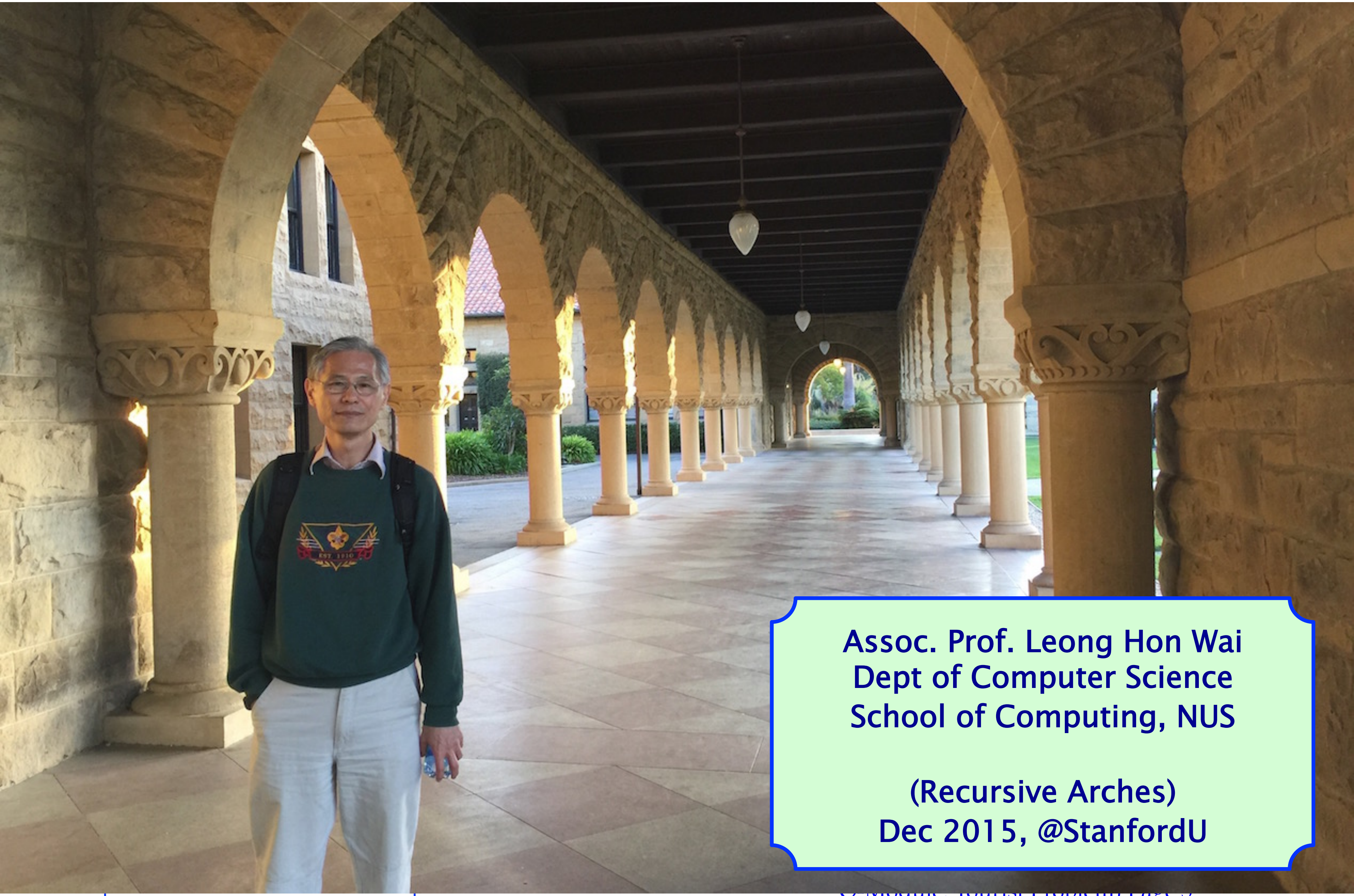
Content of Lecture Videos

❑ **Lecture 5:**

- **Problem Solving, the CS Way**
- Tourist Problem (6 short fun videos)
- Problem Transformation

❑ **Lecture 6:**

- Algorithm Design (abstraction, decomposition)
- Developing an ITeMS (IT-enabled MindSet)
- Computability and Efficiency



Assoc. Prof. Leong Hon Wai
Dept of Computer Science
School of Computing, NUS

(Recursive Arches)
Dec 2015, @StanfordU

What I *do* and *like to* do…

Research:

Design and analysis of algorithms
[算法设计与分析]

Transportation Logistics

Computational Biology
[生物信息学]

Teaching:

Algorithms Design and Analysis

Discrete Mathematics

Graph Theory and Algorithms

Problem Solving in Computing

Invitation to Computer Science

Selected Hobbies:

Mentor / Develop Talents
[SMP, SRP, UROP, etc]

National Olympiad in Informatics
[新加坡信息学奥林匹克竞赛]

International Olympiad in Informatics
[国际信息学奥林匹克竞赛]

codeXtremeApps

[_24_hr_code_jam=$fun]

[24 hours CodeJam]
since 2007

Maker Faire® Singapore

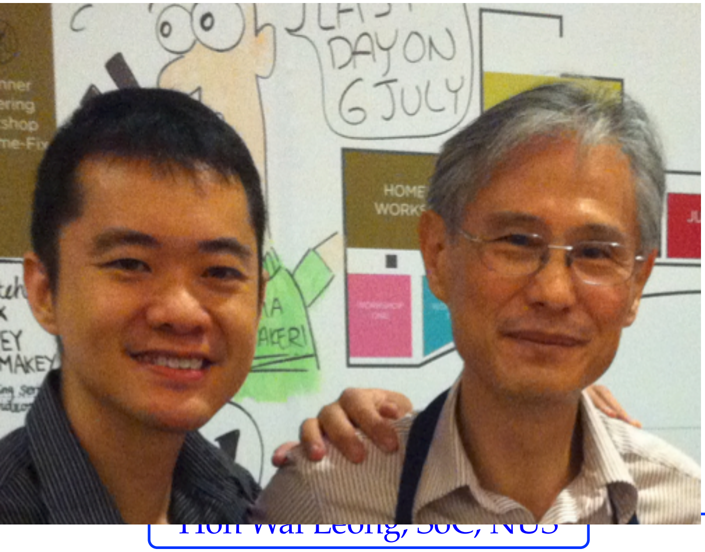
Saturday & Sunday
June 25 & 26, 2016

Singapore University of
Technology and Design

NUS
National University
of Singapore

School of Computing

Computational Thinkerer



Hon Wai Leong, SoC, NUS

Dr. Melvin Zhang &
Prof Leong Hon Wai
@Maker Faire SG
25-26 June 2016

Content

- □ Components of the CT Segment
- □ Self-Introduction
- ⇨ □ Quick intro to CS and CT
- □ CT is for Everyone
- □ Problem Solving via “Asking Questions”

Quick look at Computer Science

- A very young field
  - First computers were built only in 40’s, 50’s
- Yet, today CS is everywhere
  - Everyone uses computing devices, software
  - CS and IT has reached all areas, all sectors
- But, the fundamental questions remain the same.

Questions asked in CS…

What is the nature of problem & computation

- What problems/questions are computable?
- What problems are not computable?

Incompleteness Theorems

- There are problems that are not computable.

Search for Good Solutions…

Given a computable problem *P*,

- Can we design an algorithm to solve P?
- Can we design a faster algorithm for P?

What does it mean to have a fast algorithm?

- Is there a limit to how fast it can go?
- What is the fastest algorithm to solve P?

What about the (big) data…

Given the data for problem *P*,

- How to better organize (big) data?
- Better data analytics
- Better data visualization?

Speed and Efficiency

How to measuring “speed” of an algorithm?

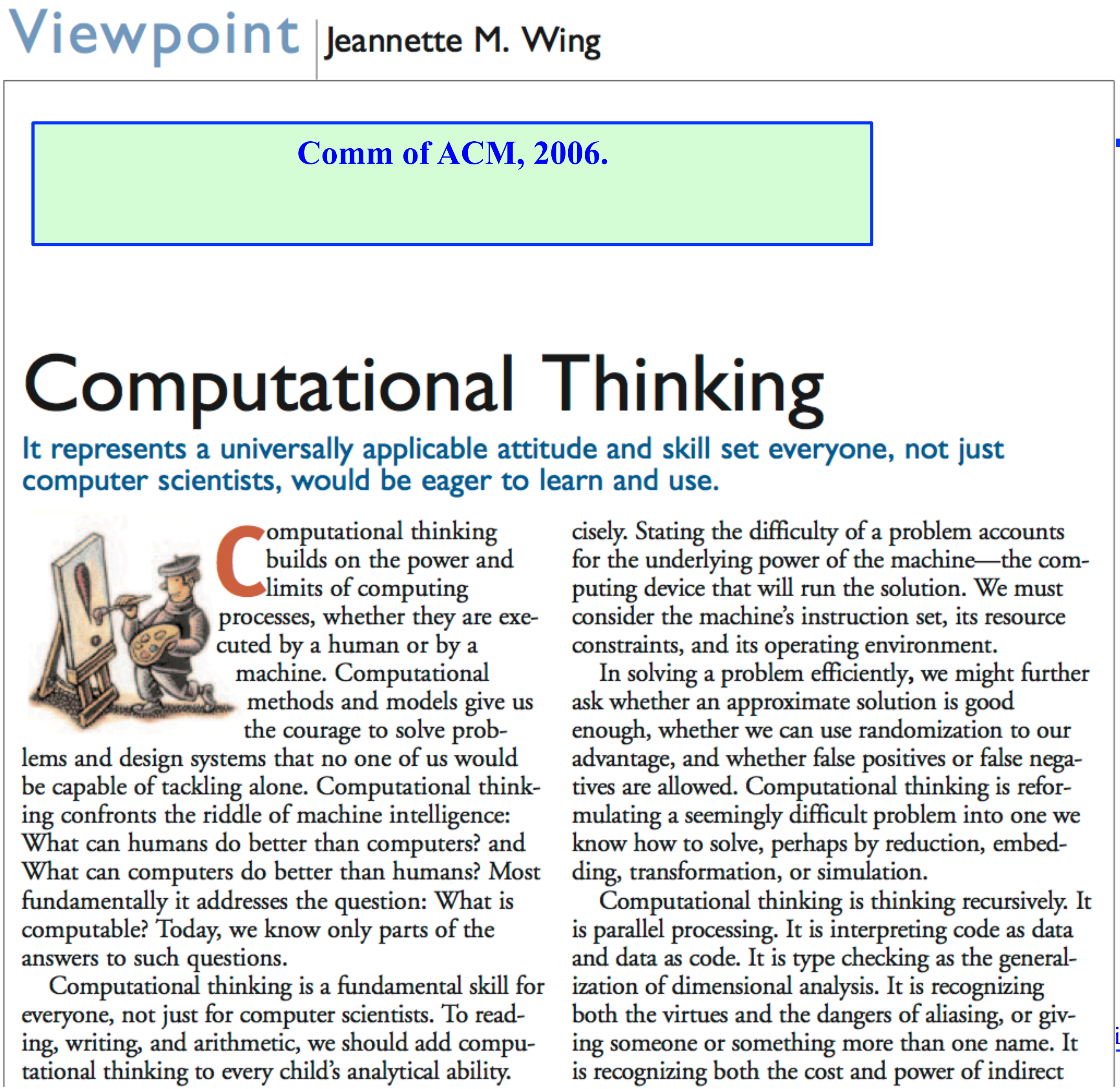
- Running time of an algorithm express in math
- Use it to compare different algorithms

Efficient algorithms vs intractable ones

- Some algorithm are correct, but “forever” to run
- They are impractical or intractable

Content

- ❑ Components of the CT Segment
- ❑ Self-Introduction
- ❑ Quick intro to CS
- ➡ ❑ CT is for Everyone
- ❑ Problem Solving via “Asking Questions”



Comm of ACM, 2006.

Computational Thinking

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.

Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of tackling alone. Computational thinking confronts the riddle of machine intelligence: What can humans do better than computers? and What can computers do better than humans? Most fundamentally it addresses the question: What is computable? Today, we know only parts of the answers to such questions.

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability.

cisely. Stating the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We must consider the machine's instruction set, its resource constraints, and its operating environment.

In solving a problem efficiently, we might further ask whether an approximate solution is good enough, whether we can use randomization to our advantage, and whether false positives or false negatives are allowed. Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation.

Computational thinking is thinking recursively. It is parallel processing. It is interpreting code as data and data as code. It is type checking as the generalization of dimensional analysis. It is recognizing both the virtues and the dangers of aliasing, or giving someone or something more than one name. It is recognizing both the cost and power of indirect

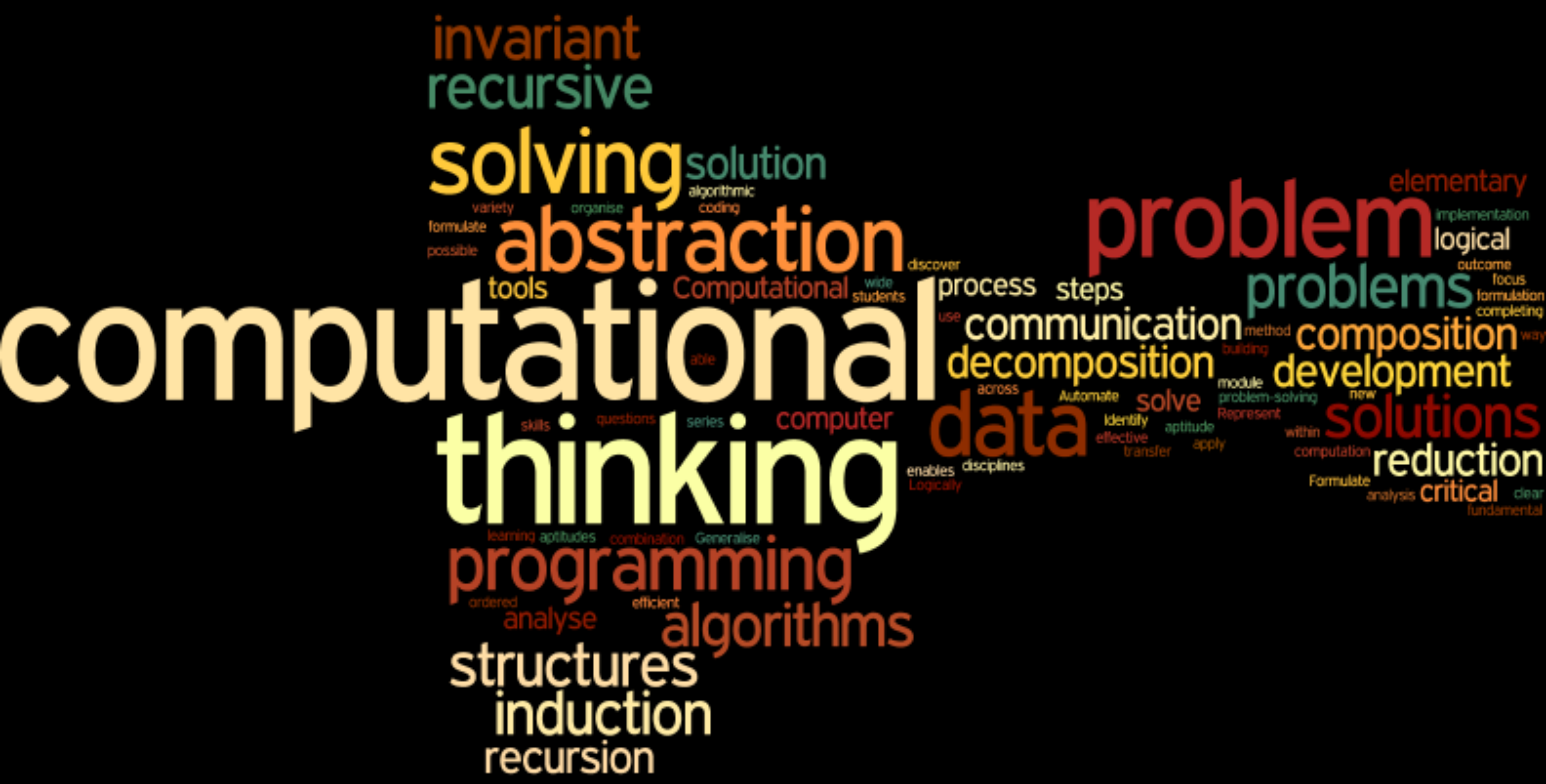
Computational Thinking…

A definition:

"Computational thinking is a way of solving problems, designing systems, and understanding human behaviour that draws on concepts fundamental to computer science. To flourish in today's world, computational thinking has to be a fundamental part of the way people think and understand the world."

(…Jeannette Wing, CMU, 2006)

A "Computational Thinking" Wordle



Computational Thinking involves

Problem Formulation

Abstraction

Algorithm Design

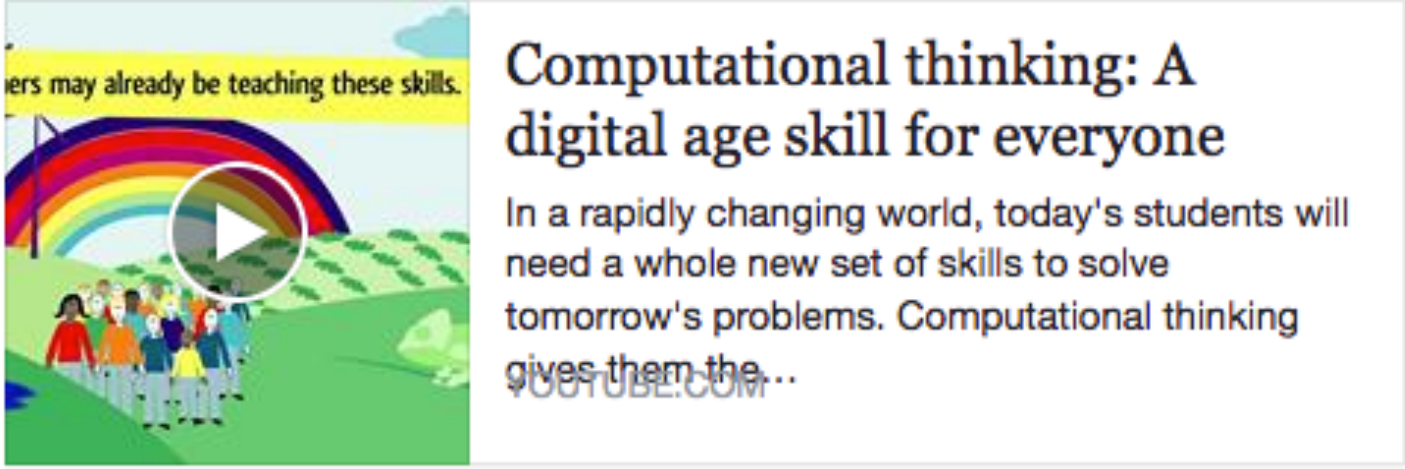
Decomposition
Composition

Finding a Pattern

Thinking Abstractly
Thinking Algorithmically
Thinking Recursively
Thinking Concurrently

Computational thinking: A digital age skill for everyone

(A 4-min video from ISTE)
https://www.youtube.com/watch?v=VFcUgSYyRPg



Computational thinking: A digital age skill for everyone

In a rapidly changing world, today's students will need a whole new set of skills to solve tomorrow's problems. Computational thinking gives them the...

YOUTUBE.COM

Develop ITeMS

Knows How to Leverage on IT
(what's possible, what's not)

ITeMS: IT Enabled MindSet
(Able to apply IT knowledge to solve problems)

"If you know *what is possible*,
you can *always* find the right people to do it."

Quick Summary:

We introduced the CT Segment

Gave quick intro to CS & CT

Argue for ITeMS

(End of video 5.0a)

Watch "Polya's 4-Step PS Process"
in Video 5.0b

If you want to contact me,

Email: leonghw@comp.nus.edu.sg

NUS
National University
of Singapore

School of Computing