DESIGN AND ANALYSIS OF ALGORITHMS

CS 4120/5120

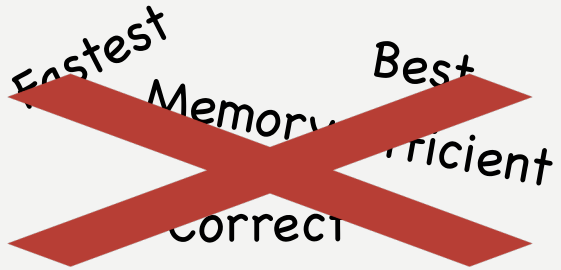
THE ROLE OF ALGORITHMS

AGENDA

- The definition of an algorithm
- Terms in developing an algorithm
- Different algorithmic problems
- Expressing algorithms

ALGORITHM?

- Various definitions
  - An algorithm is a **step-by-step** procedure which, starting with an **input** instance, produces a suitable **output**.
  - An algorithm is a well-defined **computational procedure** that takes some values as **input** and produces some values as **output**.
  - An algorithm is a **sequence of computational steps** that transform the **input** into the **output**.

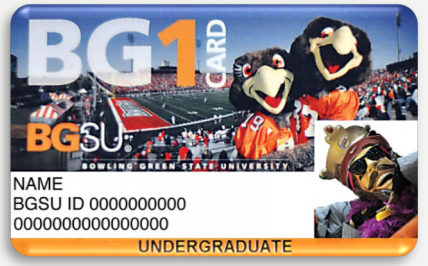


MAKING IT MORE CS
UNDERSTANDING THE PROBLEM

- The statements of the problem specifies in general terms the desired input/output relationship.

- Examples
  - Transform a pile of out-of-order nuts and bolts (left) into organized nuts and bots (right)

  - A database system that stores student records by the year of their enrollment then by their BGSU ID.



MAKING IT MORE CS
MODELING THE PROBLEM

- A computational problem can be modeled by mathematics

- Example
  - The sorting problem
    - Input: A sequence of n numbers $< a_1, a_2, \ldots, a_n >$
    - Output: A permutation (reordering) $< a'_1, a'_2, \ldots, a'_n >$ of the input sequence such that $a'_1 \leq a'_2 \leq \cdots \leq a'_n$.
  - Describe an $O(n \lg n)$ algorithm.

MAKING IT MORE CS
WORKING ON AN INSTANCE

- An ***instance of a problem***
  - Normally, we are given (or we come up with) an ***input*** sequence. Such an input sequence is called an ***instance*** of the problem.
  - The ***instance*** satisfies the constraints imposed in the problem statement.

- Example
  - An ***input*** sequence for the sorting problem: $< 31, 41, 59, 26, 41, 58 >$
  - An ***input*** sequence for the sorting problem on distinct numbers: $< 31, 41, 59, 26, 58 >$

MAKING IT MORE CS
CORRECTING THE ALGORITHM

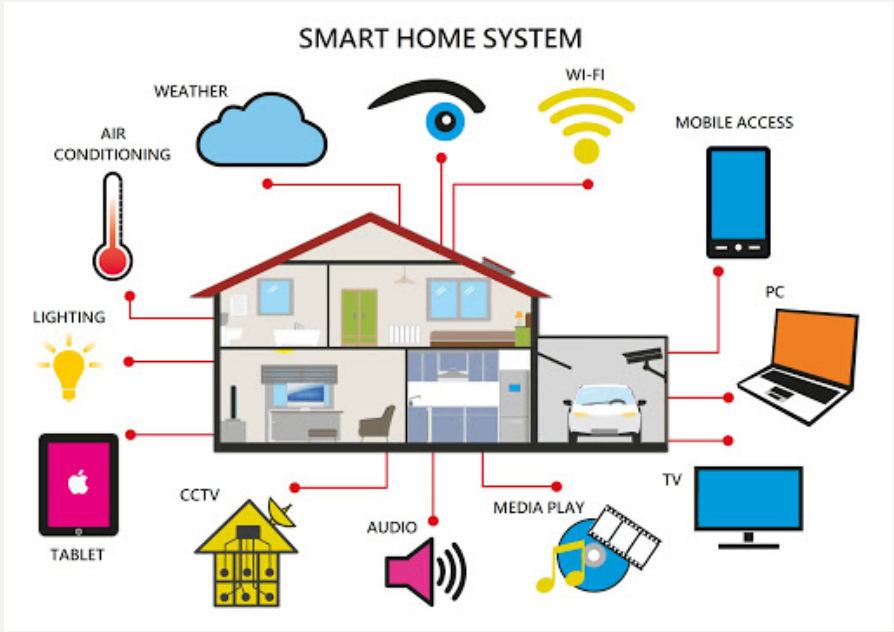
- Algorithm is said to be ***correct*** if, for every input instance, it halts with the correct output.
- We say that a correct algorithm solves the given computational problem.

ALGORITHMIC PROBLEMS
PROBLEMS WITH *BEST* SOLUTION

- The Human Genome Project

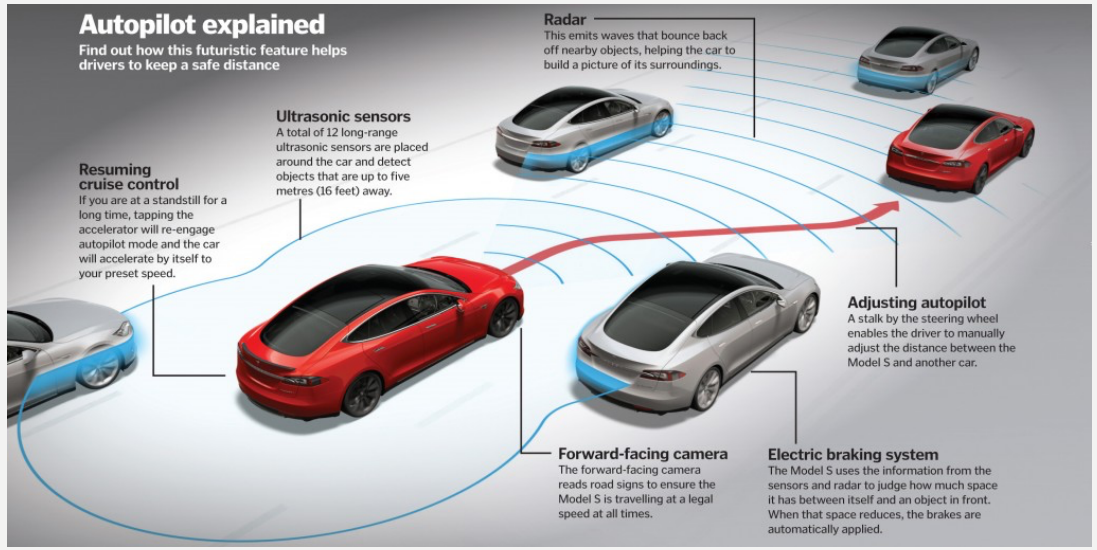
ALGORITHMIC PROBLEMS
PROBLEMS WITH *BEST* SOLUTION

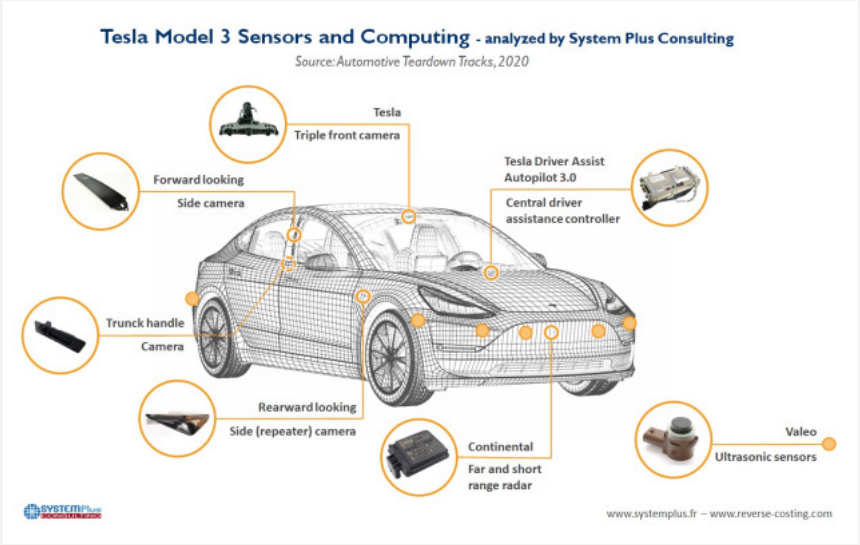
- Cyberphysical systems
  - Internet routing, networking algorithms
  - Big data
  - Simple AI system (automation)
  - Network resource allocations



ALGORITHMIC PROBLEMS
PROBLEMS WITH *BEST* SOLUTION

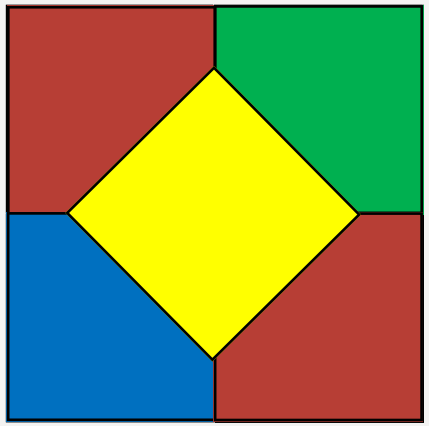
- Auto-piloting Systems
- Smart Transportation Systems

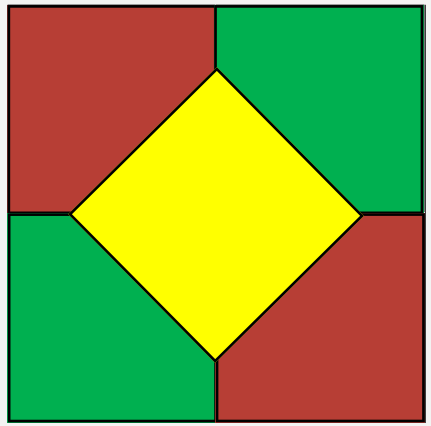




ALGORITHMIC PROBLEMS
PROBLEMS WITHOUT *BEST* SOLUTION

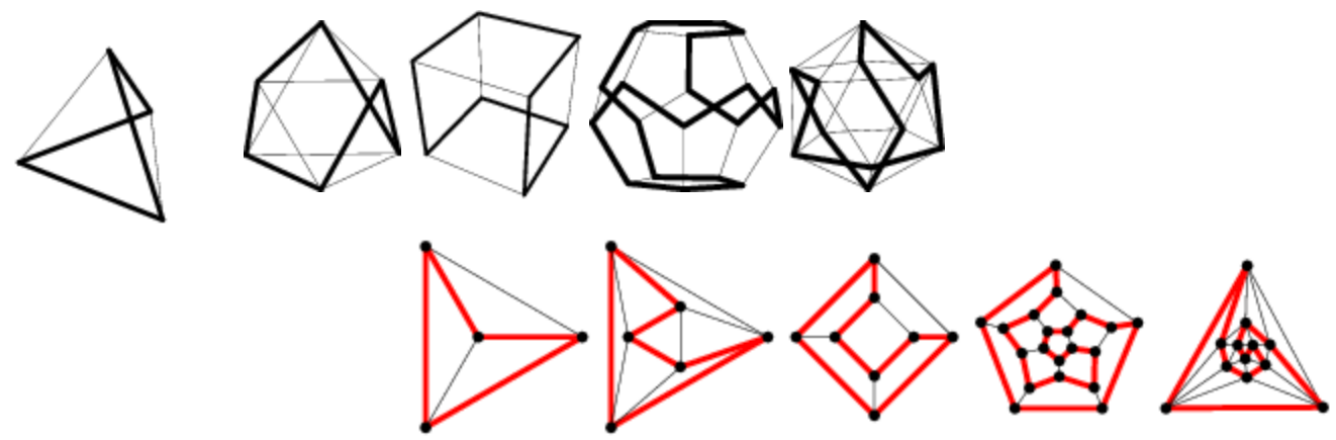
- k-Colorability
  - The four-color theorem (left, using red, yellow, green, and blue)
  - The three-colored plane (right, using red, yellow, and green)





ALGORITHMIC PROBLEMS
PROBLEMS WITHOUT *BEST* SOLUTION

- The Hamiltonian Cycle
  - A Hamiltonian cycle, also called a Hamiltonian circuit, Hamilton cycle, or Hamilton circuit, is a *graph cycle* (i.e., closed loop) through a graph that visits each node exactly ONCE.



ALGORITHM AS A TECHNOLOGY

- Choose the algorithm that works *efficiently* in terms of time and space.
  - Parallel algorithms to support parallel computing
  - Appropriate routing algorithms to support web-applications
  - Variation of an algorithm to support large-scale input size

EXPRESSING ALGORITHMS
PLAIN ENGLISH SENTENCES

- Example
  - Input: An array A of n distinct numbers.
  - Output: the largest number, or the *max*, and the smallest number, or the *min* of A.

Solution 1
Step 1: Sort A in increasing order.
Step 2: Output the first and last number in the sorted list as the min and max of the sequence.

Solution 2
Step 1: Scan array A to compute the max.
Step 2: Scan array A to compute the min.

EXPRESSING ALGORITHMS
ENGLISH + MATH EXPRESSIONS

- Example
  - Input: An array A of n distinct numbers.
  - Output: the largest number, or the *max*, and the smallest number, or the *min* of A.

Solution 3

Step 1: Initialize temporary variables $curr\_min = curr\_max = A[1]$, assuming the starting index is 1.

Step 2: For each array element $A[i]$, where $i \geq 2$, do the following.

- if $A[i] > curr\_max$, $curr\_max = A[i]$
- else if $A[i] < curr\_min$, $curr\_min = A[i]$

EXPRESSING ALGORITHMS
ENGLISH + MATH EXPRESSIONS

- Example
  - Input: An array A of n distinct numbers.
  - Output: the largest number, or the *max*, and the smallest number, or the *min* of A.

Solution 4

Step 1: Initialize two empty arrays, $SMALL$ and $LARGE$.

Step 2: Compare pairs of numbers in array A.

- For each pair, store the smaller number in $SMALL$, the larger number in $LARGE$.
- If n is odd, compare the last number with the 2nd to the last number.

Step 3: Initialize $curr\_min = SMALL[1]$, $curr\_max = LARGE[1]$.

Step 4: For all numbers in $SMALL$, if $SMALL[i] < curr\_min$, $curr\_min = SMALL[i]$.

Step 5: For all numbers in $LARGE$, if $LARGE[i] > curr\_max$, $curr\_max = LARGE[i]$.

EXPRESSING ALGORITHMS
ENGLISH + MATH + COMPLEXITY

- Example
  - Input: An array A of n distinct numbers.
  - Output: the largest number, or the *max*, and the smallest number, or the *min* of A.
  - Objective: Design an algorithm that uses as few comparisons as possible.

- Take notes of the four solutions, participate in your breakout room to determine the number of comparisons in terms of n. (10 minutes)
  - Hint: you may determine the number of comparisons depending on the parity of n.
  - Hint: you may consider using $\lceil \quad \rceil$ or $\lfloor \quad \rfloor$ operator.

EXPRESSING ALGORITHMS
PSEUDOCODE

- In this class, we shall typically describe algorithms as programs written in a ***pseudocode***
  - Similar to C, C++, Java, Python, or Pascal
  - The most clear and concise expressive method to specify an algorithm
    - Contain codes that resemble "real" program code
    - Could contain English sentences

```
INSERTION-SORT(A)
1    for j = 2 to A.length
2         key = A[j]
3         // Insert A[j] into the sorted sequence A[1..j-1]
4         i = j – 1
5         while i > 0 and A[i] > key
6              A[i+1] = A[i]
7              i = i – 1
8         A[i+1] = key
```

NEXT UP
PSEUDOCODE CONVENTIONS

- Table of references
  - https://rb.gy/wsp0wo
  - https://rb.gy/tysgus
  - https://rb.gy/bcirdy
  - https://rb.gy/gugbix
  - http://visioforce.com/smarthome.html
  - http://www.onelectriccars.com/how-does-teslas-autopilot-mode-work/1401/
  - https://www.eetasia.com/teslas-hardware-retrofits-for-model-3/