CMPS 10 9-24-09

Chapter 1: Introduction

What is Computer Science? Some possible misconceptions are:

- The study of computers
- Programming and programming languages
- Applications software like MS word, Adobe Photoshop, etc.

Computer Science is the study of Algorithms, especially their

(1) Mathematical Properties: Correctness, Efficiency, Complexity (time and space)

(2) Hardware Realizations: Logic Gates, Circuits, Architecture

(3) Software Realizations: Programming and Programming Methodologies

(4) Applications to Other Disciplines: Mathematics, Physics, Engineering, Business

Ex.

$$\begin{array}{r} 1\ 1\ 0\ 0 \\ 4\ 9\ 3 \\ +\ 7\ 5\ 1 \\ \hline 1\ 2\ 4\ 4 \end{array}$$

Ex.

$$\begin{array}{r} 1\ 0\ 1\ 0 \\ 6\ 1\ 7 \\ +\ 9\ 4\ 5 \\ \hline 1\ 5\ 6\ 2 \end{array}$$

$m = 3$

$a_2 = 6,\ a_1 = 1,\ a_0 = 7$

$b_2 = 9,\ b_1 = 4,\ b_0 = 5$

$$945 = 9 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$$

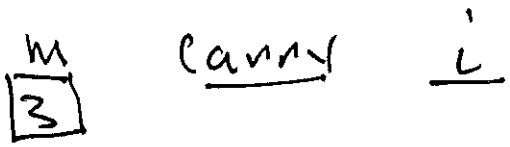
$$= b_2 \cdot 10^2 + b_1 \cdot 10^1 + b_0 \cdot 10^0$$

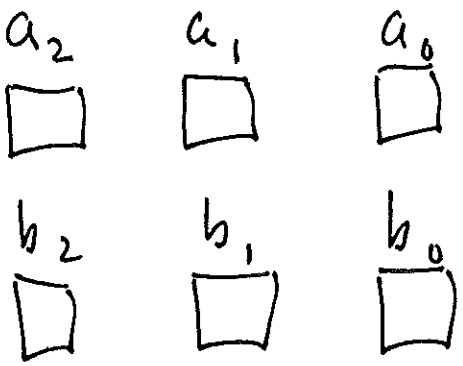
Problem: add two m-digit numbers, where $m \geq 1$.

Input: $m \geq 1$ and two m-digit numbers: $a_{m-1}a_{m-2}\ldots a_2a_1a_0$ and $b_{m-1}b_{m-2}\ldots b_2b_1b_0$

Output: Their sum: $c_m c_{m-1}c_{m-2}\ldots c_2c_1c_0$

```
1)  carry←0
2)  i←0
3)  while i<m do 4-10
4)      c_i ← a_i + b_i + carry
5)      if c_i ≥10 do 6-7
6)          c_i ← c_i −10
7)          carry←1
8)      else do 9
9)          carry←0
10)     i←i+1
11) c_m ←carry
12) print c_m c_{m-1} c_{m-2} ... c_2 c_1 c_0
13) stop
```





m
3

a_2	a_1	a_0
6	1	7

b_2	b_1	b_0
9	4	5

Carry	1
~~0~~	~~0~~
+	+
~~0~~	~~2~~
1	3

c_3	c_2	c_1	c_0
1	~~15~~ 5	6	~~12~~ 2

1	5	6	2

EXERCISE

TRACE THIS ALGORITHM ON

$m = 4$

a_3	a_2	a_1	a_0
9	3	7	1

b_3	b_2	b_1	b_0
8	5	2	0

Note that the above operations can be classified into three types:

(1) Sequential Perform a single task, then move to the next operation in the list.

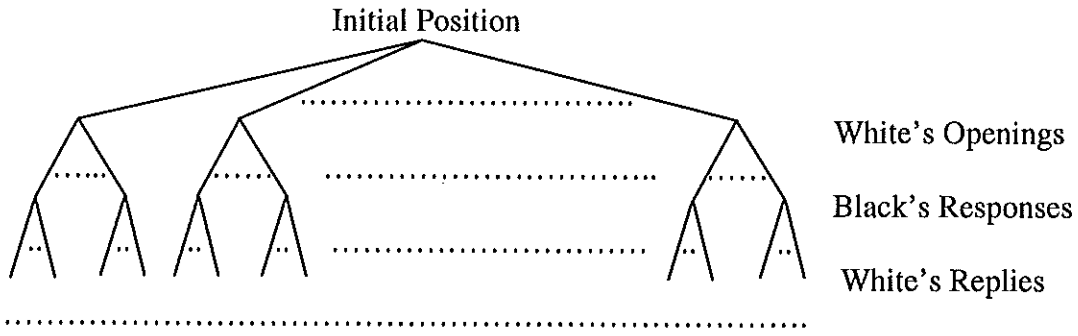
(2) Conditional (Branching) Select the next operation based value of a logical expression.

(3) Iterative (Loop) Repeat some block of instructions until some condition is met.

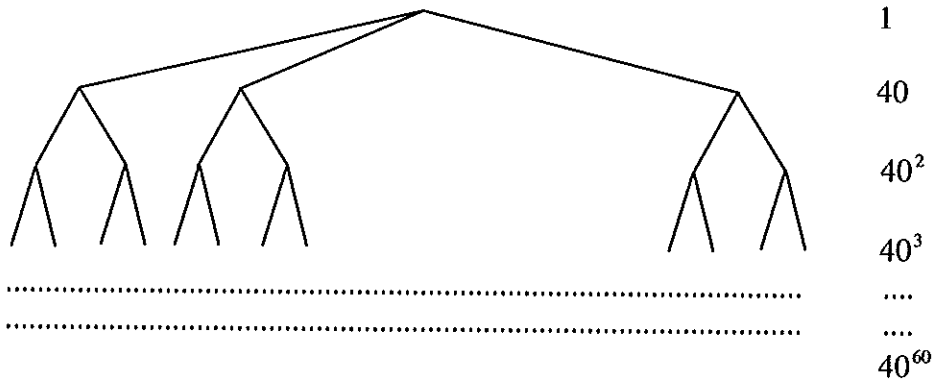
The entity (machine, robot, person, or persons) which executes the instructions in an algorithm is called the Computing Agent.

There are problems for which we have algorithmic solutions, but these algorithms are so inefficient as to be unusable.

Example Brute force chess analysis



- On average there are 40 legal moves from any board position.
- On average a game of chess takes about 30 moves (i.e. 30 moves for White and 30 moves for Black, so 60 plys.)
- Therefore there are approximately 40 board positions at depth 1 in the tree, 40^2 positions at depth 2, and in general 40^k board positions at depth k.
- Since the average game has 60 plys, the algorithm must check approximately $40^{60} \approx 10^{96}$ ending positions to determine whether they are win for White, win for Black, or Draw.



Suppose the algorithm can evaluate 1 quintillion = 10^{18} board positions per second (a ridiculously high number.) The run time of the algorithm would then be

$$\text{Time} = \frac{10^{96} \text{ positions}}{10^{18} \text{ positions/second}} = 10^{78} \text{ seconds} \approx 10^{70} \text{ years}$$

The present age of the universe is estimated by cosmologists to be about 10 billion $= 10^{10}$ years.