

ends ID 1-10-08

Ex.

$w$   
3

$a_2 \ a_1 \ a_0$   
6 | 1 | 7

$b_2 \ b_1 \ b_0$   
9 | 4 | 5

$c_3 \ c_2 \ c_1 \ c_0$   
1 | 5 | 6 | 2

→ → → →  
1 5 6 2

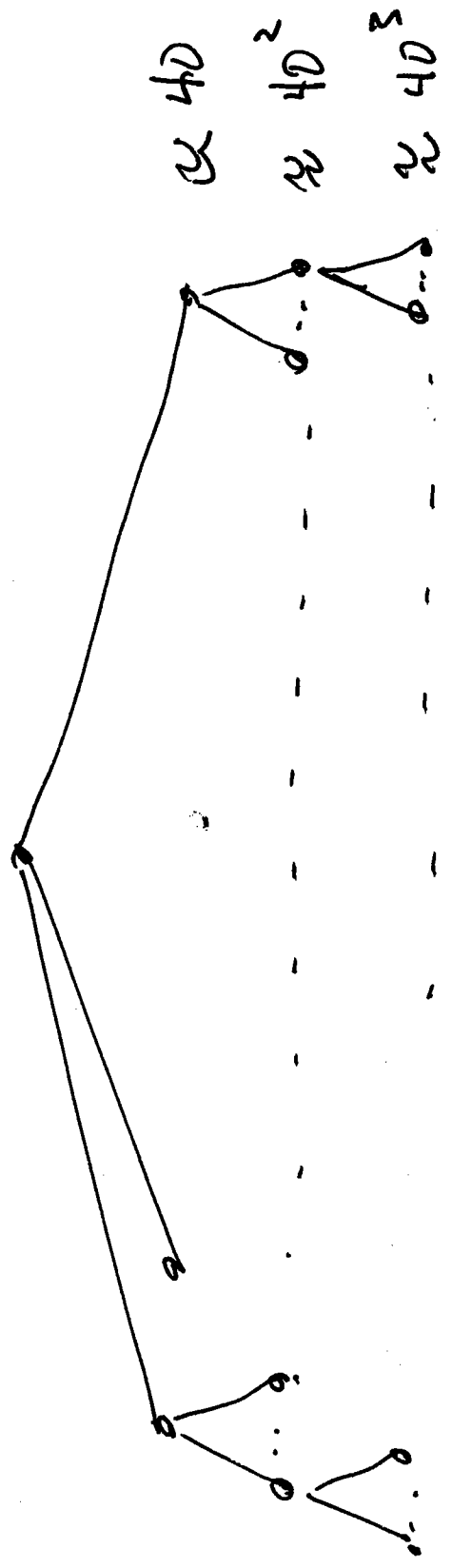
carry  
 $\phi + \phi -$

$\phi + \phi -$

Types of operations:

- (1) SEQUENTIAL: PERFORM A TASK, GO TO NEXT IN THE LIST
- (2) CONDITIONAL (BRANCHING): SELECT NEXT EXPRESSION BASED ON TRUTH VALUE OF A LOGICAL EXPRESSION
- (3) ITERATIVE (LOOPING): REPEAT SOME BLOCK OF STATEMENTS UNTIL SOME CONDITION BECOMES TRUE OR FALSE.





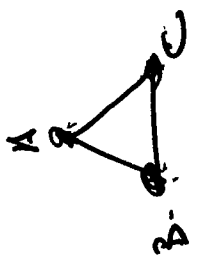
$\approx 4D^{60}$

NOTE  $4D^{60} \approx 10^{96}$

ASSUME CAN EVALUATE 1 QUANTUM POS/SEC =  $10^{18}$  POS/SEC

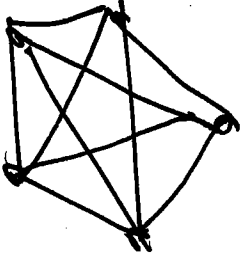
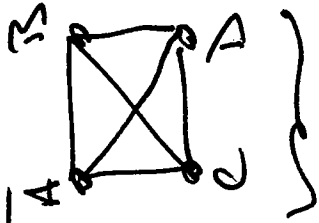
$$\text{TIME} = \frac{10^{96} \text{ POS}}{10^{18} \text{ POS/SEC}} = 10^{78} \text{ SEC} \approx 10^{70} \text{ YEARS}$$

TRAVELING SALESMAN PROBLEM!



- ABC
- ACB
- BAC
- BCA
- CAB
- CBA

6



Formal Defn. of Algorithm

An Algorithm is a well ordered collection of unambiguous and effectively comparable operations which, when executed, produce a result and halts in finite time.

Well ordered: Always know what to do next

Ex (not well ordered)

- 1.) do something
  - 2.) do something
  - 3.) do something
  - 4.) repeat
- Repeat what? ↙

UNAMBIGUOUS : Any op. which can be carried out directly by the computing algorithm. Also called Primitive operation.

Effectively Computable : Done by some computing

Algorithm.

EX.  $a \leq b$  NOT Effectively Computable (ie b < 0)

EX. LIST ALL PRIME NUMBERS  
2, 3, 5, 7, 11, 13, 17, ...

Producers A Result :

Halts in Finite Time : Only Finitely Many

Operations will be performed.

Ex. (Does not halt)

- 1.) do something
- 2.) do something
- 3.) go to (1)

} Infinite loop



# Chap 2. Algorithm Design

DETAILED & EXACT

EXPRESSIVE & ABSTRACT



COMPUTER  
LANGUAGES

PSEUDO-CODE

NATURAL  
LANGUAGES

• SEQUENTIAL OPS.

- COMPUTATION

- INPUT

- OUTPUT

COMPUTATION & ASSIGNMENT :

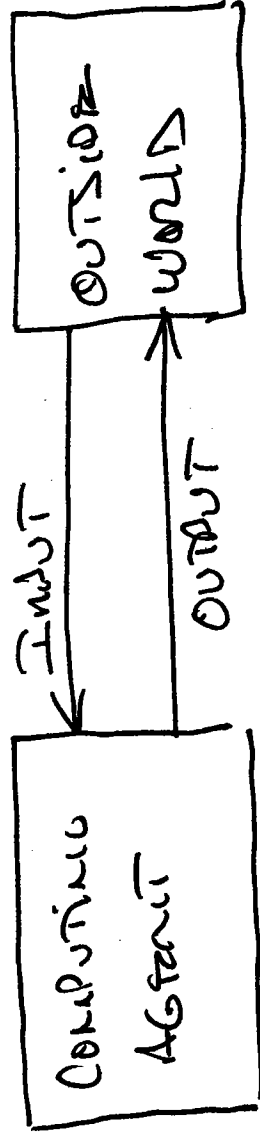
variable ← expression

↑ ASSIGNMENT OPERATOR

Ex  $c_i \leftarrow a_i + b_i + \text{Carry}$

Ex.  $x \leftarrow \frac{-b + \sqrt{b^2 - 4ac}}{2a}$

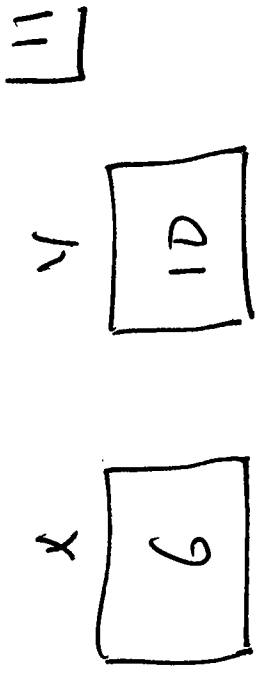
INPUT  $\rightarrow$  OUTPUT :



get variable

print variable,

print ~~expression~~



```
get x
print x
```

```
print x+1
```

```
print 'hello'
```

```
print 'error: Division by zero'
```

```
print 'the value of x is'
```

```
print x
```

```
print ' and the value of y is'
```

```
print y
```

Ex

Ex

Ex. A straight line Algorithm

- 1.) get  $a, b, c, d$
- 2.)  $sum \leftarrow a + b + c + d$
- 3.)  $average \leftarrow \frac{sum}{4}$
- 4.) print average
- 5.) stop

a	b	c	d
2	1	3	-1

↑

FOR EXAMPLE

Sum
5

Average
1.25

CONJUNCTION OPERATIONS

⋮

if condition

do something	}	INVERSE
do something		
do something		

do something

⋮

if condition

[do something  
:  
do something]

TRUE  
→

else

[do something  
:  
do something]

FALSE  
→

do something  
:  
: