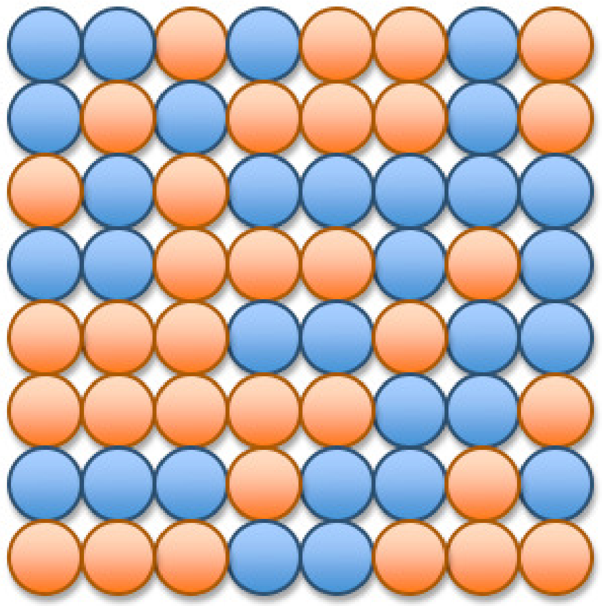
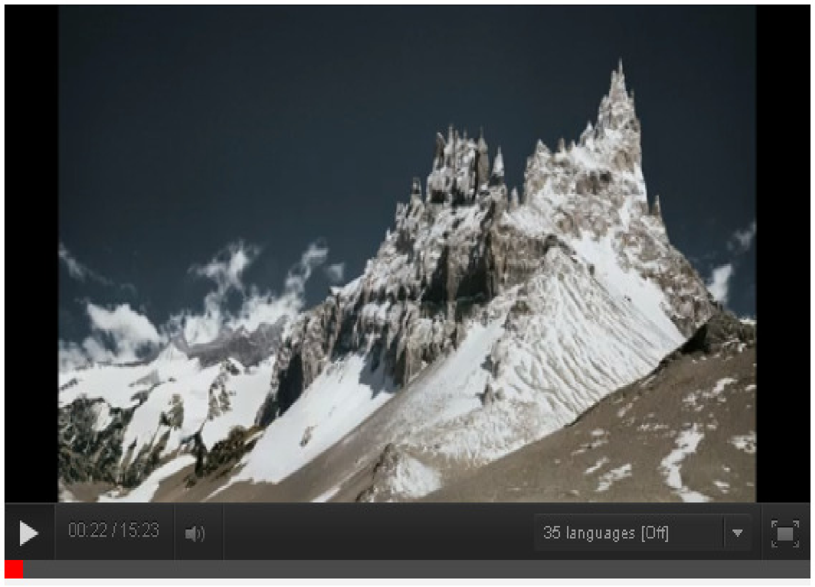


UNIT 3A
Algorithmic Thinking

Algorithms Rule the world

Kevin Slavin: How algorithms shape our world

FILMED JUL 2011 • POSTED JUL 2011 • TEDGlobal 2011



Kevin Slavin argues that we're living in a world designed for -- and increasingly controlled by -- algorithms. In this riveting talk from TEDGlobal, he shows how these complex computer programs determine: espionage tactics, stock prices, movie scripts, and architecture. And he warns that we are writing code we can't understand, with implications we can't control.

source- TED

Algorithms

- An algorithm is “a precise rule (or set of rules) specifying how to solve some problem.” (thefreedictionary.com)
- The study of algorithms is one of the key foundations of computer science.

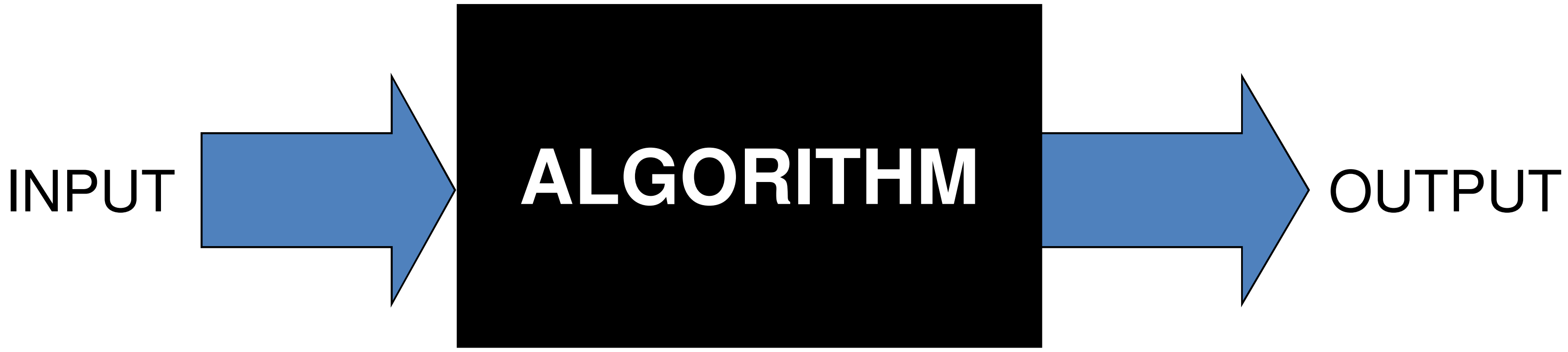
How to Change a flat tire is an Algorithm

- *Arrange the following set of instructions as a <u>correct program</u> to change a flat tire*
- *Instructions*
  - *Lift the car* → 6
  - *take the old tire out* – 8
  - *Stop the car* → 2
  - *Lower the car* – 12
  - *Take the tools out* → 3
  - *Place the hub cap* – 11
  - *Drive the car* – 14
  - *Choose a good spot* → 1
  - *Put the spare tire* – 9
  - *loosen the lug nuts* – 5
  - *Remove the flat tire* – 8
  - *Remove the hub cap* – 4
  - *Put the tools and old tire back in car* – 13
  - *Put the lug nuts back in* – 10
  - take off nuts – 7

Instruction set

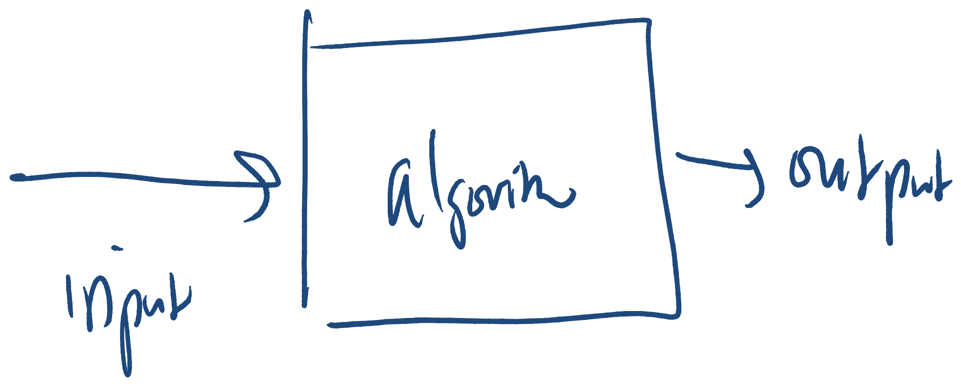
An algorithm is like a function

$$F(x) = y$$



A Program is an implementation of an algorithm

- Program components
  - Input
  - Algorithm
  - output



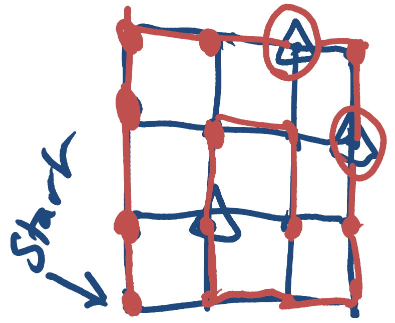
Input

- Input specification
    - What the program needs
        - How much data is required?
        - What kind of data is required?
        - In what form will this data be received by the algorithm?

Eg: Changing a tire:

Computation done according to an algorithm

- An algorithm requires clear and precisely stated steps that express how to perform the operations to yield the desired results.
- Algorithms assume a basic set of primitive operations that are assumed to be understood by the executor of the algorithm.
  - Robot – start, stop, turn right, move a block, pick a cone, place a marker



Output

```
def add(x,y)
   return x+y
end
```

```
add(add(2,3),add(4,5))
```

- Output specification
  - Specify the type of output expected
    - A print out
    - A return value
    - A return list etc
- Output specification for computational algorithms:
  - What results are required?
  - How should these results be reported?
  - What happens if no results can be computed due to an error in the input? What do we output to indicate this?

What makes a “good” algorithm?

- A good algorithm should produce the correct outputs for any set of legal inputs.
- A good algorithm should execute efficiently with the fewest number of steps as possible.
- A good algorithm should be designed in such a way that others will be able to understand it and modify it to specify solutions to additional problems.

$m=2,\ n=3$ ✓

$m=2,\ n=0$ ✓

$m=2,\ n=-1$

```
sum = 0
for i in 1..n
    sum = sum + m
end
return sum
```

Act out an algorithm

- Sorting a random list of students
  - Input
  - output

- Finding the maximum of a random set of students
  - Input
  - output

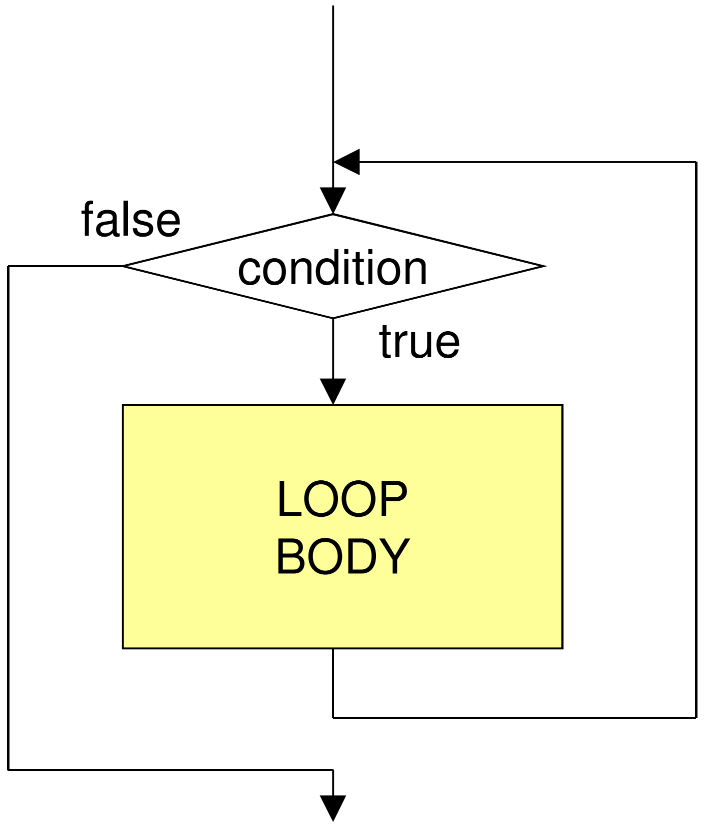
`while` loop

Format:

```
while condition do
    loop body
end
```

one or more instructions to be repeated

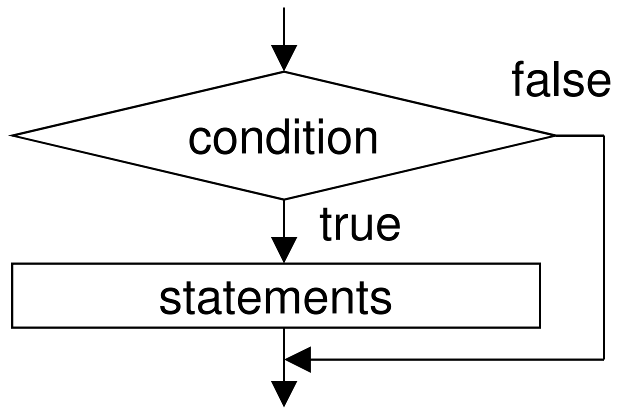
If the loop condition becomes false during the loop body, the loop body still runs to completion before we exit the loop and go on with the next step.



`if` statement

Format:

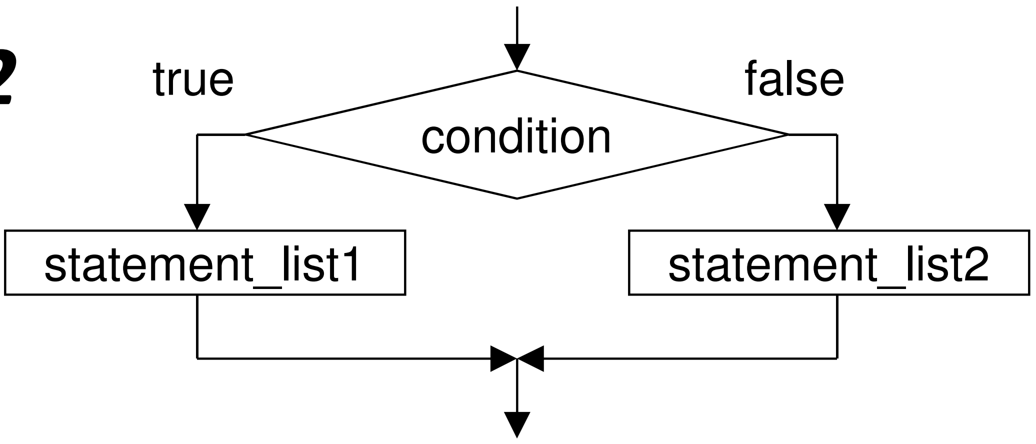
```
if condition then
    statement_list
end
```



if/else statement

Format:

```
if condition then
    statement_list1
else
    statement_list2
end
```



Think about these algorithms

- Checking whether a number is prime or not
- Printing all primes up to some n
- Finding the largest prime?
- Finding a path through a maze
- Writing a tic tac toe player
- Writing a program to find closest friend to you
- Writing a program to find the all 3-degrees of separation