

CSH1F2 Introduction to Computer Science

Week 2: Algorithm

Author: Lecturer Team

Bachelor of Engineering, School of Computing



Outline

1. Algorithm

ALGORITHM

- In daily life, an algorithm is a set of steps or instructions that can be followed to complete a task.
- If we can express a task as an algorithm, then we can give those instructions to someone else (or a computer) to take care of that task for us.
- Real-Life Examples of Algorithms
 - Cooking: Recipe for preparing a dish
 - Origami: The Art of Paper Folding
 - Directions: How to go to Husein Sastra Airport

Algorithm for Computer Program

- ▶ Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages.
- ▶ People write programs to execute algorithms.
- ▶ Before write programs we must express algorithm in pseudocode
- ▶ See this : 'What is an algorithm' Video and 'BBC Learning'

Basic Building Blocks of Algorithm

Selection, Iteration, and Sequencing are building blocks of algorithms. Every algorithm can be constructed using only sequencing, selection, and iteration.

- **Sequencing:** also known as “action” - is the application of each step of an algorithm in the order in which the statements are given.
- **Selection:** also known as “decision” or “branching” – most commonly seen in if-statements . It gives us a way to compare two things (numbers) and take action if one thing was true.
- **Iteration:** also known as “repetition” or “loop” – the repetition of part of an algorithm until a condition is met or for a specified number of times.

Sequence / Action Example

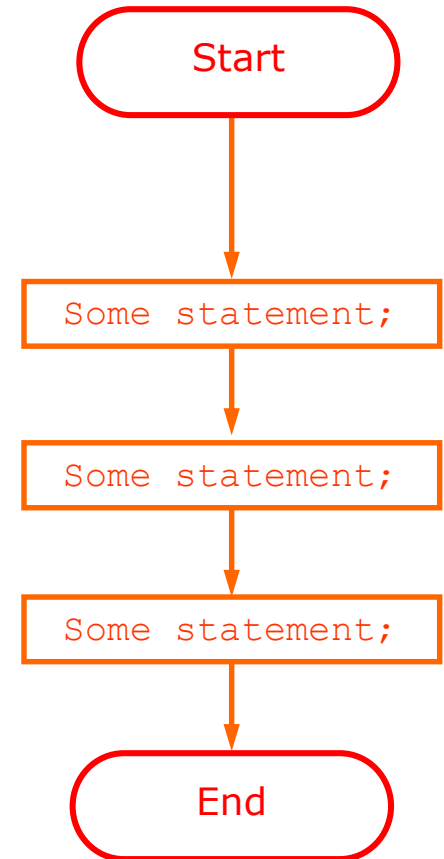
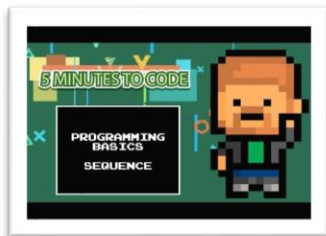
read x

read y

write x

write y

See this :



Selection / Decision Example

read x

read y

if $x > y$

then write x

else write y

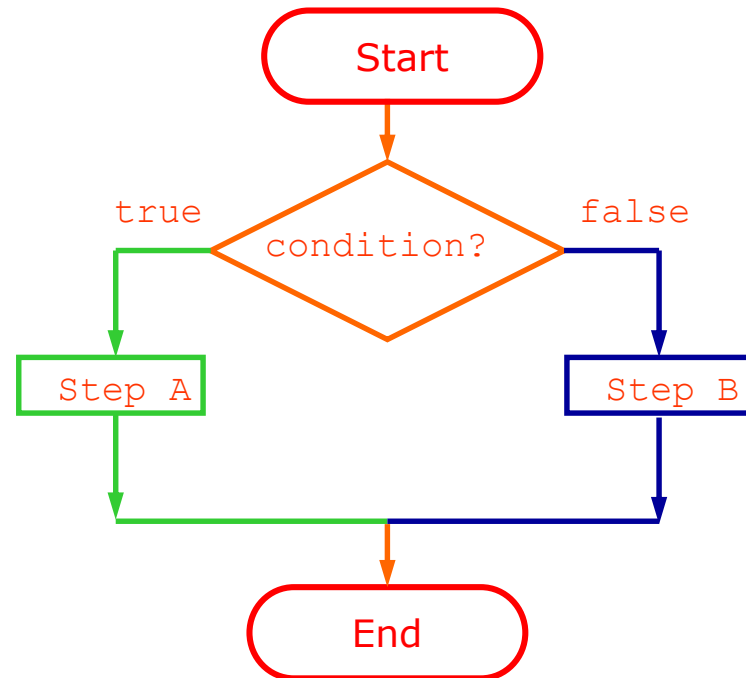
See this :



A Decision is making a choice among several actions.

Example:

If you listen to KMEL radio, print "Friend of the Camel", otherwise print "Don't like Camels."



Iteration / Repetition Example

count \leftarrow 1

while count \leq 5

 read x

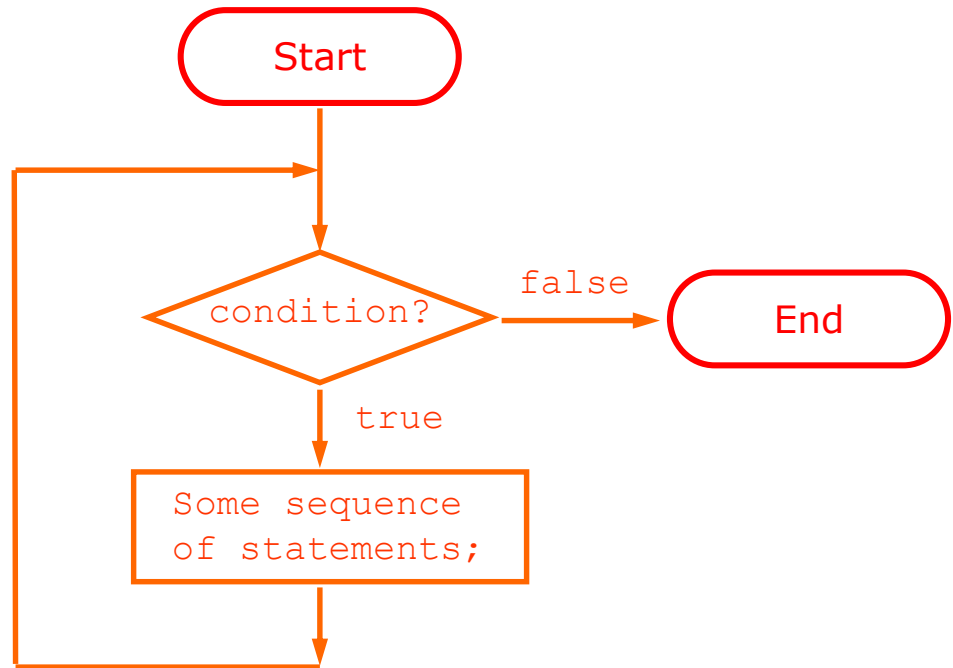
 read y

 if (x > y)

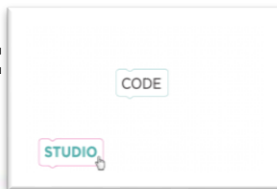
 then write x

 else write y

 count \leftarrow count + 1



See this :



A Repetition is one or more instructions that the computer performs repeatedly.
Example: Print "GOOD GRIEF, CHARLIE BROWN" 77 times.

Algorithms for problem solving

- **1. Different algorithms can be developed to solve the same problem**
- Creativity in algorithm comes from figuring out clever ways to solve problems by developing a process that could be executed by a machine.
- **2. Different programs (code) can be written to implement the same algorithm.**
- Describe the features of a programming language that make it different from the language you are used to using in everyday life. Explain why a programming language must be created in this way.

Algorithm Design

- ▶ The methods of algorithm design form one of the core practical technologies of computer science.
- ▶ Algorithms can be combined to make a new algorithms

Some important algorithms to learn

This list is subjective, there are other similar opinions out there which turn out to give us highlights on what algorithms are often employed.

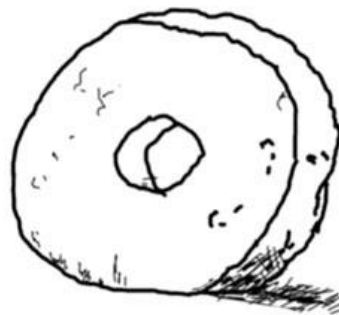
1. Sort algorithms
2. Search algorithms
3. Hashing
4. Dynamic programming
5. Exponentiation by squaring
6. String matching and parsing
7. Primality testing algorithm
8. Error handling algorithm

Algorithm and Version Control

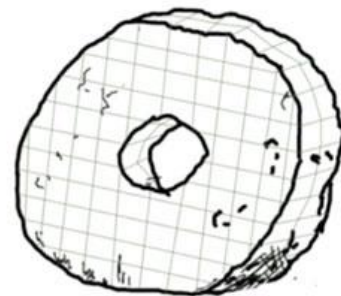
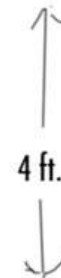
Algorithms are created by collaboration and evolve, hence this makes history rich. We often need to review history to retain and improve our knowledge ...
“not reinventing the wheel”

Check this out:

<https://guides.github.com/introduction/git-handbook/>



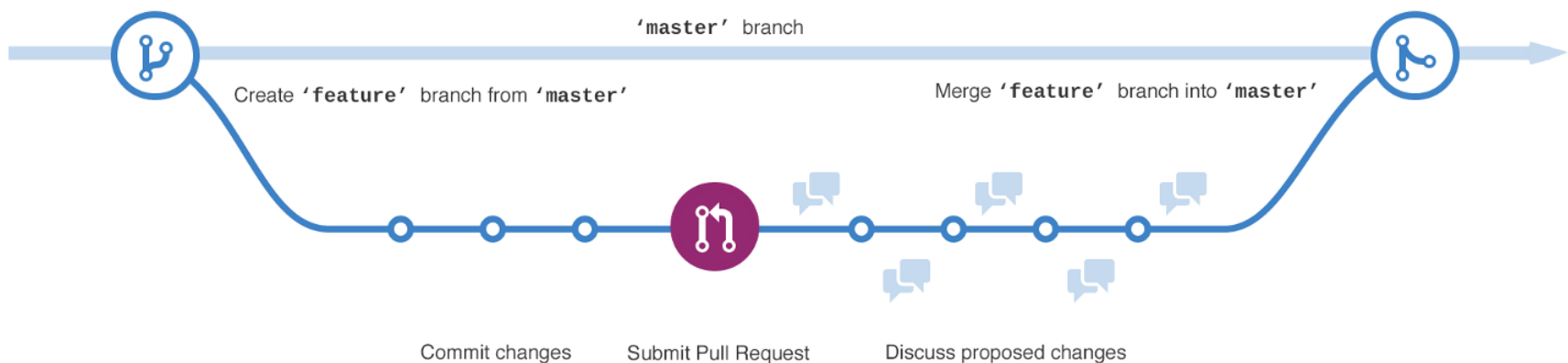
1. Prehistoric wheel



2. 3D Computer Model

The Github Flow

- ▶ Anything in the “master branch” is “deployable”
- ▶ Works must be done in separate branches other than “master”
- ▶ “Pull request” (in purple) is a way of collaboration



CREATING FUNCTION

- ▶ A form of abstraction that enables programmer to write code in larger, more logical chunks and focus on what something does, rather than how something does.
- ▶ Function must be defined and called in program.
- ▶ Benefit of using function
 - programs become easier to read and write
 - functions remove the need to repeat large chunks of code
 - functions break the program into logical chunks

Function Example

```
1 drawStep();  
2  
3 function drawStep() {  
4   moveForward();  
5   turnLeft();  
6   moveForward();  
7   function right() {  
8     turnLeft();  
9     turnLeft();  
10    turnLeft();  
11  }  
12 }
```

- ▶ There are 2 functions defined here: `drawStep()` and `right()`
- ▶ Can you guess:
Why function `right()` doesn't work?

TOP DOWN DESIGN

- ▶ Use “top-down” problem-solving approach to identify sub-tasks of a larger programming task
- ▶ It is similar to “Divide and Conquer” approach

VOCABULARY

- ▶ **Abstraction** - Pulling out specific differences to make one solution work for multiple problems.
- ▶ **Function** - A piece of code that you can easily call over and over again

Class Project - Sample

- Explain what kind of algorithm in the case below

PROBLEM : What is the largest integer
INPUT: All integers { ... -2, -1, 0, 1, 2, ... }
OUTPUT: The largest integer
Algorithm: Arrange all the integers in a list in decreasing order;
 MAX = first number in the list;
 Print out MAX;

- Write two different algorithms to solve multiplication of two integer numbers x and y

Class Project - Session 2

Note:

If you have trouble adding remote repository, you can check this link :

<https://help.github.com/articles/creating-a-personal-access-token-for-the-command-line/>

<https://stackoverflow.com/questions/37937984/git-refusing-to-merge-unrelated-histories-on-rebase>

Substitute any words including the bracket <...> with related instruction.

Instructions:

1. Make a directory in your computer OS called CSH1F2-2018-IF42INT
2. Change to the directory
3. Initiate a local repository in the directory by issuing this command : `git init`
4. Check the status of your repository with this : `git status`
5. Check the commit history in your repository with this : `git log --oneline`
6. Add remote :
 1. IF42INT `git remote add origin https://github.com/herutomo/CSH1F2-2018-IF42INT.git`
 2. IF4202 `git remote add origin https://github.com/herutomo/CSH1F2-2018-IF4202.git`
7. Check the list of remote repository with this : `git remote -v`
8. Pull content of remote repository from master branch : `git pull origin master`
9. Create your branch in local repository dan enter to the branch : `git checkout -b <your_student_id>`
10. Submit Quiz file with these series of command :
 - a. Answer all questions in `Class-Project-Session-2-1.txt` and rename it to `Class-Project-Session-2-1-<your_student_id>.txt`
 - b. Add the file : `git add Class-Project-Session-2-1-<your_student_id>.txt`
 - c. Commit the file : `git commit Class-Project-Session-2-1-<your_student_id>.txt -m "<your_full_name> <your_student_id>"`
 - d. Update to remote repository : `git push origin <your_student_id>`



THANK YOU