MUST

Wisdom & Virtue

MIRPUR UNIVERSITY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF SOFTWARE ENGINEERING

Design & Analysis of Algorithm

(Lecture # 3)

Algorithm & Flow chart

Saba Zafar
(Lecturer)

Date: 9-5-2023

Algorithm

Algorithm

An algorithm is defined as **sequence of steps to solve a problem (task)**. The steps must be finite, well defined and unambiguous. Writing algorithm requires some thinking. Algorithm can also be defined as a plan to solve a problem and represents its logic. Note that an algorithm is of no use if it does not help us arrive at the desired solution

Algorithm characteristics

- **It should have finite number of steps**. No one can be expected to execute infinite number of steps.
- The steps must be in order and simple
- **Each step should be defined clearly i.e. without un-ambiguity (without doubtfulness)**
- Must include all required information
- Should exhibit at least one output

Flowchart

1. **Flowchart**
2. **A flowchart is a pictorial (graphical) representation of an algorithm**. A flowchart is drawn using different kinds of symbols. A symbol is used for a specific purpose. Each symbol has name.

Algorithm	Flowchart	Program
An algorithm is defined as **sequence of steps to solve a problem (task)**.	**A flowchart is pictorial (graphical) representation of an algorithm**.	Set of instructions. Instruction is a command to the computer to do some task.
Algorithm can also be defined as a plan to solve a problem and represents its logic.	A picture is worth of 1000 words. We can understand more from picture than words.	Implementation of Algorithm or flowchart

CONT!!!

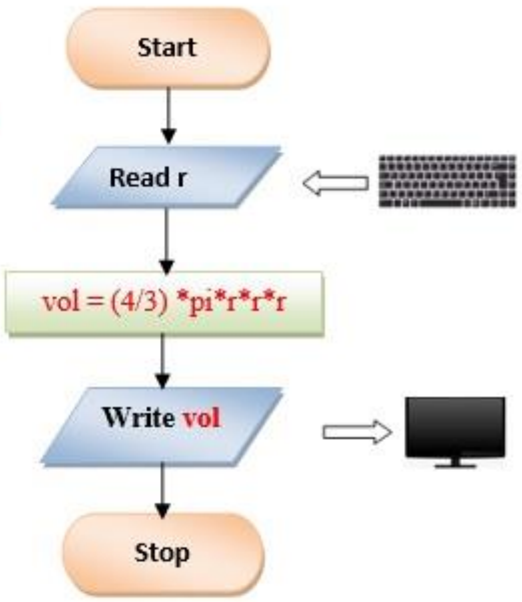
1. Different algorithms have different performance characteristics to solve the same problem. Some algorithms are fast. Some are slow. Some occupy more memory space. Some occupy less memory space. Some are complex and some algorithms are simple.
2. **Logically algorithm, flowchart and program are the same.**
3. **Q1**. Create a program to compute the volume of a sphere. Use the formula: $V = (4/3) * pi * r^3$ where pi is equal to 3.1416 approximately. The **r** is the radius of sphere. Display the result.

Example

Algorithm

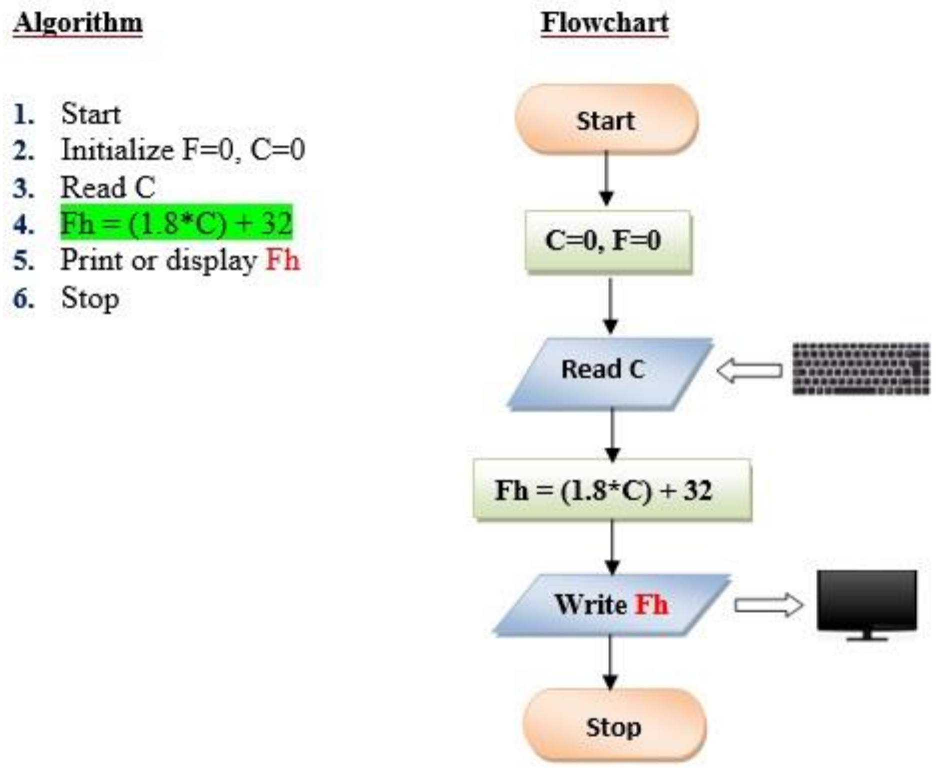
1. Start
2. Read r
3. vol = (4/3) *pi*r*r*r
4. Print or display vol
5. Stop

Flowchart



Example

1. . Write a program the converts the input Celsius degree into its equivalent Fahrenheit degree. Use the formula: F = (9/5) *C+32.



Algorithm

1. Start
2. Initialize F=0, C=0
3. Read C
4. Fh = (1.8*C) + 32
5. Print or display Fh
6. Stop

Flowchart

Start

C=0, F=0

Read C

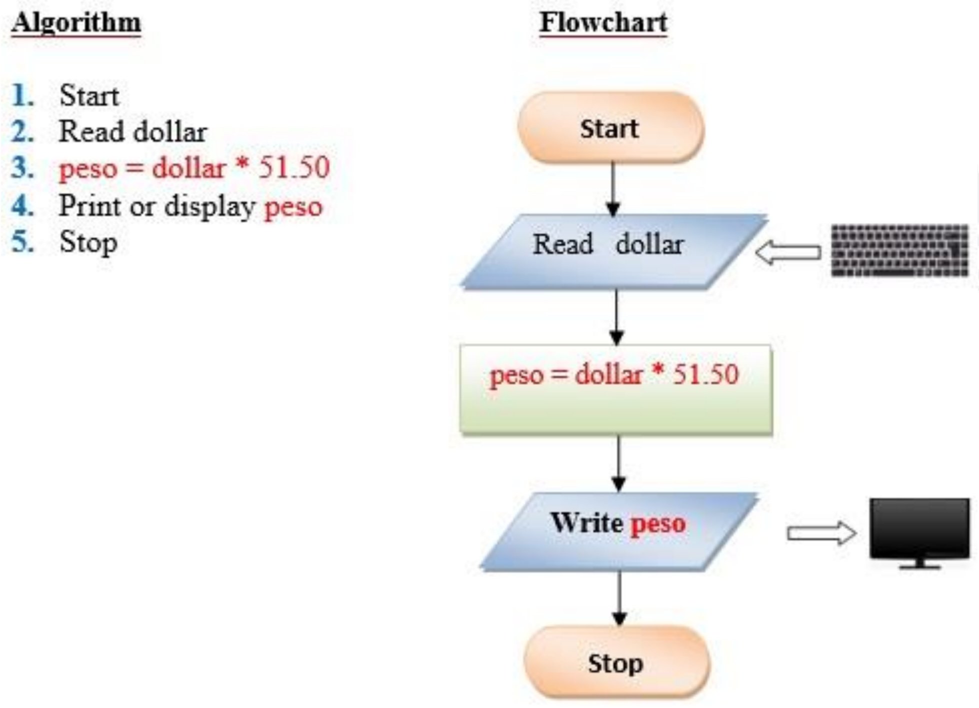
Fh = (1.8*C) + 32

Write Fh

Stop

Example

1. Write a program that converts the input dollar to its peso exchange rate equivalent. Assume that the present exchange rate is 51.50 pesos against the dollar. Then display the peso equivalent exchange rate.



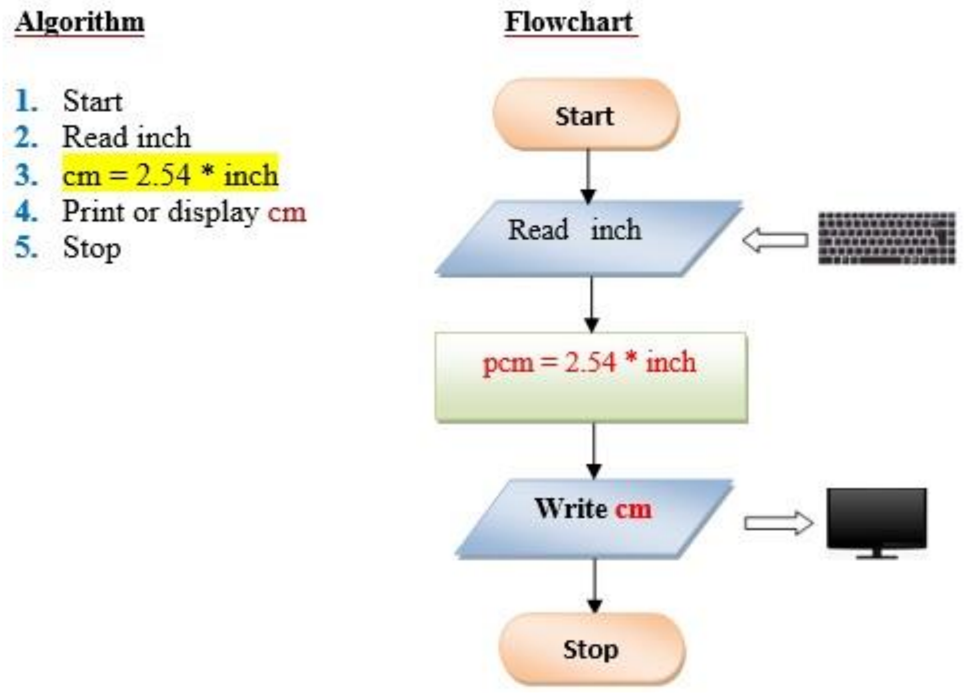
Algorithm

1. Start
2. Read dollar
3. peso = dollar * 51.50
4. Print or display peso
5. Stop

Flowchart

Example

Write a program that converts an input inch(es) into its equivalent centimeters. Take note that one inch is equivalent to 2.54cms.



Algorithm

1. Start
2. Read inch
3. cm = 2.54 * inch
4. Print or display cm
5. Stop

Flowchart

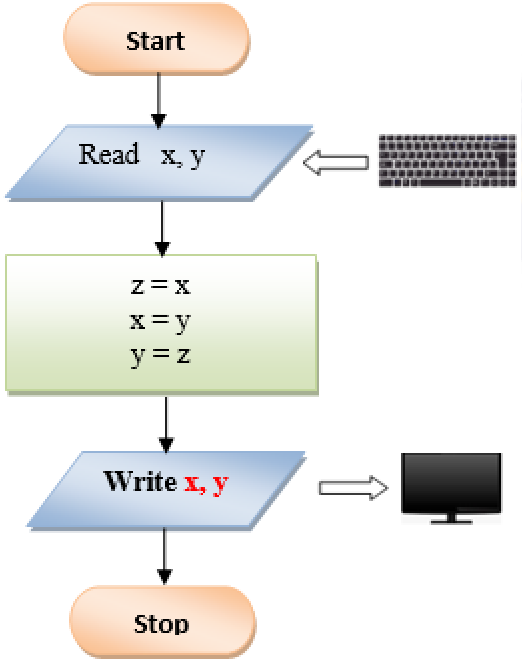
Example

- Write a program that exchanges the value of two variables: x and y. The output must be: the value of variable y will become the value of variable x, and vice versa.

Algorithm

```
1. Start
2. Read x, y
3. Declare third variable, z
           z = x
           x = y
           y = z
4. Print or display x, y
5. Stop
```

Flowchart



Algorithm Design Techniques

Algorithm Design Techniques

The following is a list of several popular design approaches:

1. Divide and Conquer Approach: It is a top-down approach. The algorithms which follow the divide & conquer techniques involve three steps:

Divide the original problem into a set of subproblems.

Solve every subproblem individually, recursively.

Combine the solution of the subproblems (top level) into a solution of the whole original problem.

Greedy Technique: Greedy method is used to solve the optimization problem. An optimization problem is one in which we are given a set of input values, which are required either to be maximized or minimized (known as objective), i.e. some constraints or conditions.

Greedy Algorithm always makes the choice (greedy criteria) looks best at the moment, to optimize a given objective.

The greedy algorithm doesn't always guarantee the optimal solution however it generally produces a solution that is very close in value to the optimal.

CONT!!!

- **Dynamic Programming:** Dynamic Programming is a bottom-up approach we solve all possible small problems and then combine them to obtain solutions for bigger problems.
- This is particularly helpful when the number of copying subproblems is exponentially large. Dynamic Programming is frequently related to **Optimization Problems**.
- **4. Branch and Bound:** In Branch & Bound algorithm a given subproblem, which cannot be bounded, has to be divided into at least two new restricted subproblems. Branch and Bound algorithm are methods for global optimization in non-convex problems. Branch and Bound algorithms can be slow, however in the worst case they require effort that grows exponentially with problem size, but in some cases we are lucky, and the method coverage with much less effort.

CONT!!!

Randomized Algorithms: A randomized algorithm is defined as an algorithm that is allowed to access a source of independent, unbiased random bits, and it is then allowed to use these random bits to influence its computation.

6. Backtracking Algorithm: Backtracking Algorithm tries each possibility until they find the right one. It is a depth-first search of the set of possible solution. During the search, if an alternative doesn't work, then backtrack to the choice point, the place which presented different alternatives, and tries the next alternative.

7. Randomized Algorithm: A randomized algorithm uses a random number at least once during the computation make a decision.

Divide and Conquer Introduction

Divide and Conquer Introduction

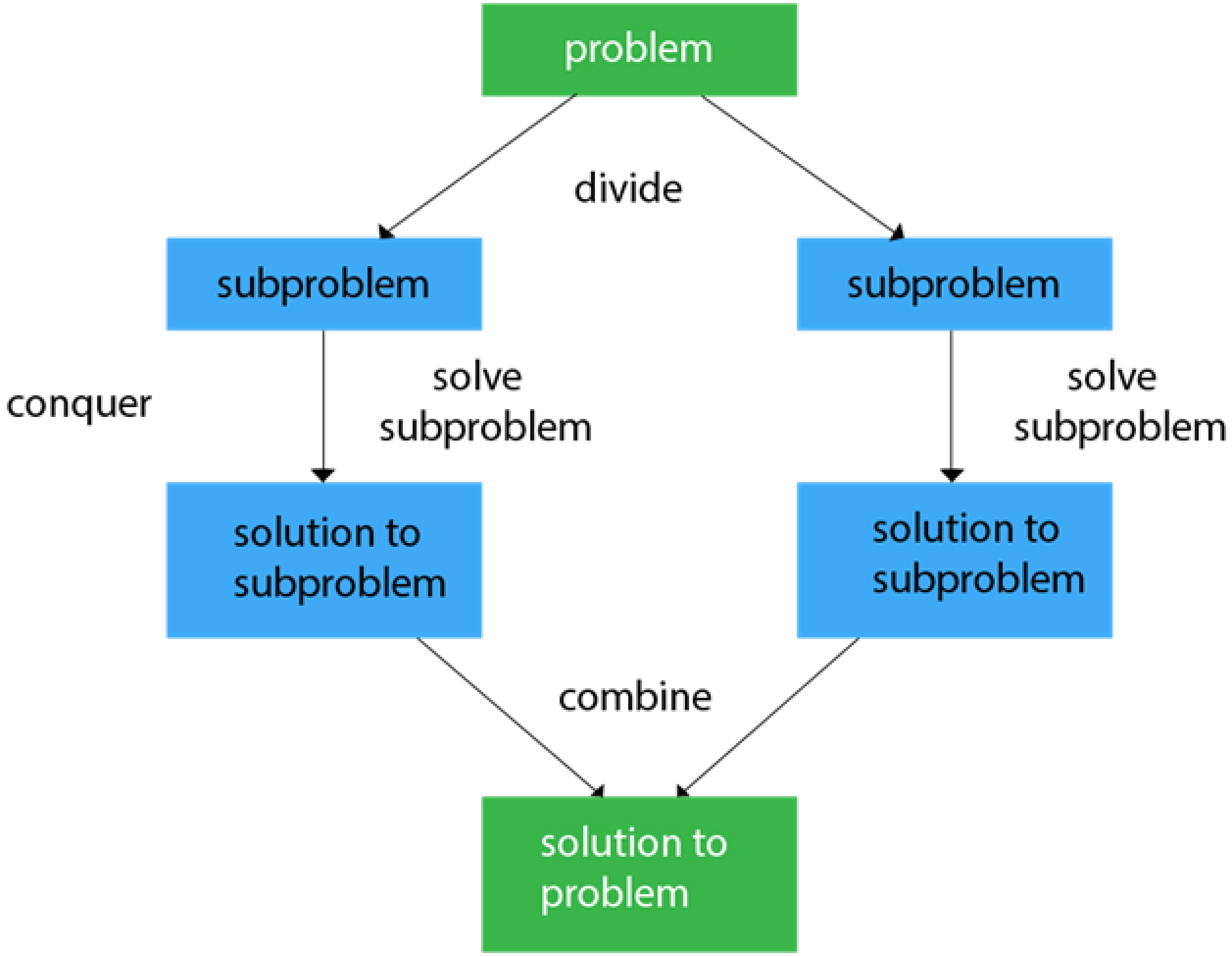
Divide and Conquer is an algorithmic pattern. In algorithmic methods, the design is to take a dispute on a huge input, break the input into minor pieces, decide the problem on each of the small pieces, and then merge the piecewise solutions into a global solution. This mechanism of solving the problem is called the Divide & Conquer Strategy.

Divide and Conquer algorithm consists of a dispute using the following three steps.

Divide the original problem into a set of subproblems.

Conquer: Solve every subproblem individually, recursively.

CONT!!!


problem
divide
subproblem
subproblem
conquer
solve subproblem
solve subproblem
solution to subproblem
solution to subproblem
combine
solution to problem

Example

Generally, we can follow the **divide-and-conquer** approach in a three-step process.

Examples: The specific computer algorithms are based on the Divide & Conquer approach:

Maximum and Minimum Problem

Binary Search

Sorting (merge sort, quick sort)

Tower of Hanoi.

Fundamental of Divide & Conquer Strategy:

There are two fundamental of Divide & Conquer Strategy:

Relational Formula

Stopping Condition

1. Relational Formula: It is the formula that we generate from the given technique. After generation of Formula we apply D&C Strategy, i.e. we break the problem recursively & solve the broken subproblems.

2. Stopping Condition: When we break the problem using Divide & Conquer Strategy, then we need to know that for how much time, we need to apply divide & Conquer. So the condition where the need to stop our recursion steps of D&C is called as Stopping Condition.

Applications of Divide and Conquer Approach:

- Applications of Divide and Conquer Approach:
- Following algorithms are based on the concept of the Divide and Conquer Technique:
- **Binary Search:** The binary search algorithm is a searching algorithm, which is also called a half-interval search or logarithmic search. It works by comparing the target value with the middle element existing in a sorted array. After making the comparison, if the value differs, then the half that cannot contain the target will eventually eliminate, followed by continuing the search on the other half. We will again consider the middle element and compare it with the target value. The process keeps on repeating until the target value is met. If we found the other half to be empty after ending the search, then it can be concluded that the target is not present in the array.

CONT!!!

Quicksort: It is the most efficient sorting algorithm, which is also known as partition-exchange sort. It starts by selecting a pivot value from an array followed by dividing the rest of the array elements into two sub-arrays. The partition is made by comparing each of the elements with the pivot value. It compares whether the element holds a greater value or lesser value than the pivot and then sort the arrays recursively.

Merge Sort: It is a sorting algorithm that sorts an array by making comparisons. It starts by dividing an array into sub-array and then recursively sorts each of them. After the sorting is done, it merges them back.

CONT!!!

Closest Pair of Points: It is a problem of computational geometry. This algorithm emphasizes finding out the closest pair of points in a metric space, given n points, such that the distance between the pair of points should be minimal.

Strassen's Algorithm: It is an algorithm for matrix multiplication, which is named after Volker Strassen. It has proven to be much faster than the traditional algorithm when works on large matrices.

Cooley-Tukey Fast Fourier Transform (FFT) algorithm: The Fast Fourier Transform algorithm is named after J. W. Cooley and John Turkey. It follows the Divide and Conquer Approach and imposes a complexity of O(nlogn).

Advantages of Divide and Conquer

7.Karatsuba algorithm for fast multiplication: It is one of the fastest multiplication algorithms of the traditional time, invented by Anatoly Karatsuba in late 1960 and got published in 1962. It multiplies two n-digit numbers in such a way by reducing it to at most single-digit.

Advantages of Divide and Conquer

Divide and Conquer tend to successfully solve one of the biggest problems, such as the Tower of Hanoi, a mathematical puzzle. It is challenging to solve complicated problems for which you have no basic idea, but with the help of the divide and conquer approach, it has lessened the effort as it works on dividing the main problem into two halves and then solve them recursively. This algorithm is much faster than other algorithms.

It efficiently uses cache memory without occupying much space because it solves simple subproblems within the cache memory instead of accessing the slower main memory.

It is more proficient than that of its counterpart Brute Force technique.

Disadvantages of Divide and Conquer

Since these algorithms inhibit parallelism, it does not involve any modification and is handled by systems incorporating parallel processing.

Disadvantages of Divide and Conquer

Since most of its algorithms are designed by incorporating recursion, so it necessitates high memory management.

An explicit stack may overuse the space.

It may even crash the system if the recursion is performed rigorously greater than the stack present in the CPU.

THANKS