Algorithms, pseudocode, programs

Intro. to Computing

`https://mandarm.github.io/computing.html`

Algorithm

What is an algorithm?

An algorithm is a step-by-step procedure to accomplish some objective.

Examples

- How to withdraw money from an ATM.
- How to play the Indian National Anthem on a computer.
- How to obtain the distribution of blood groups in this class.
- How to compute the square root of a positive integer.

Properties

- Typically written in natural language, e.g., English
- Must be *sufficiently detailed and unambiguous* for the intended person

Pseudocode

What is pseudocode?

Also a step-by-step procedure to accomplish some objective, but looks somewhat like a computer program

Properties

- Preferably limited to *primitives* (simple, elementary actions)
- Uses some standard programming constructs in a language-independent way
  - *conditionals:* check whether a condition holds in order to determine what to do
  - *loops:* do something repeatedly a fixed number of times, or until some criteria are fulfilled
- **BUT** no strict rules regarding the format
- Must be *sufficiently detailed and unambiguous* for the intended person

Examples: problem 1

Problem 1

Consider a reservoir, fitted with a set of taps and drains. You are given the capacity of the reservoir, and details about the taps and drains attached to the reservoir. Determine whether a partially full reservoir ever becomes empty or full.

Examples: problem 1

Problem 1

Consider a reservoir, fitted with a set of taps and drains. You are given the capacity of the reservoir, and details about the taps and drains attached to the reservoir. Determine whether a partially full reservoir ever becomes empty or full.

Input:

- L capacity of the reservoir in litres
- T number of taps attached to the reservoir
- D number of drains attached to the reservoir
- t_i time in minutes that it takes to fill an initially empty reservoir if all drains stopped, and only the i-th tap is opened $(1 \leq i \leq T)$
- d_j time in minutes that it takes to drain an initially full reservoir if all taps shut off, and only the j-th drain is opened $(1 \leq j \leq D)$

Examples: problem 1

Pseudocode

I Initially, set variables X and Y to 0.

II **for** i in $\{1, 2, 3, \ldots, T\}$
$X \longleftarrow X +$ _____.

III **for** j in $\{1, 2, 3, \ldots, D\}$
$Y \longleftarrow Y +$ _____.

IV **If** X _____ Y
then print "*A partially empty reservoir never fills up if all taps and drains are kept open.*"
else print "*An initially empty reservoir fills up in* _____ *minutes if all taps and drains are kept open.*"

V **If** X _____ Y
then print "*A partially full reservoir never becomes empty if all taps and drains are kept open.*"
else print "*An initially full reservoir becomes empty in* _____ *minutes if all taps and drains are kept open.*"

Examples: problem 2

Problem 3

A merchant buys impure salt from N suppliers $S_1, S_2, \ldots, S_N$, refines it, and sells the purified salt (all impurities are extracted and discarded) at Rs. c/kg. Fill in the blanks (marked as (i)-(v)) appropriately in the algorithm given below, so that it determines the value of c for which this entire transaction results in a total profit of Rs. X for the merchant.

Inputs: Three arrays $P[\,]$, $Q[\,]$, and $U[\,]$, each containing N elements, indexed from 1 to N.

- $P[i]$ denotes the purity of the salt bought from S_i. It is a number between 0 and 1, with 1 denoting completely pure salt.
- $Q[i]$ denotes the weight (in kg) of impure salt purchased from S_i.
- $U[i]$ denotes the cost price (per kg) of impure salt purchased from S_i.

Examples: problem 2

Pseudocode

I Initialise $CostPrice \leftarrow 0,\ Weight \leftarrow 0,\ i \leftarrow 1.$

II While _____(i)_____

(a) $CostPrice \leftarrow CostPrice +$ _____(ii)_____

(b) $Weight \leftarrow Weight +$ _____(iii)_____

(c) $i \leftarrow$ _____(iv)_____

III Output $c =$ _____(v)_____ $\longleftarrow$ This is often called the *return value*.

Examples: problem 3

Problem 3

Compute the GCD of two integers a and b.

Pseudocode

I Let r be the remainder when a is divided b.

II If r is 0, required GCD (or *return value*) is b.

III Otherwise?

Corner case: a or b can be 0 or 1

Algorithms/pseudocode to programs

What do we need?

- **Variables:** to store values, and to use them in calculations
  - variables have (dynamic) types
- **Operators** (and functions):
  - arithmetic operators
  - relational operators
  - logical operators
  - string operators / functions
- **Statements:** executing actions
  - assignment statements
  - conditional statements
  - loops
- **Input and output methods**
- **Built in structures:** lists, dictionaries, . . .