E2011: Theoretical fundamentals of computer science
Introduction to algorithms – part II

Vlad Popovici, Ph.D.

RECETOX

Outline

1. Review of pseudocode constructs
2. Basic data structures
3. Exercises

Pseudocode

- *variables* - store some values (e.g. x, y); may refer to simple (e.g. scalar) values, or more complicated *data structures* (vectors, matrices, lists, etc.)
- *input* to specify the required values for the algorithm to compute the *output*
- variables are assigned values: $x \leftarrow 50$ or $x \leftarrow y$, but values are never assigned variables or other values: $50 \leftarrow x$ is a nonsense
- mathematical operators can be used as usual

Branching - conditional execution

if ⟨*condition*⟩ **then**
    code for ⟨*condition*⟩ is True
    [
else
    code for ⟨*condition*⟩ is False
    ]
end if

- **"else"** branch is optional
- one can use **"continue"** to force quitting a loop or **"next"** to force jumping to the next iteration within a loop

Loops

Repeat as long as the condition is true:

while ⟨*condition*⟩ **do**
    instruction
    ...
end while

Repeat as long as the condition is false:

repeat
    instruction
    ...
until ⟨*condition*⟩

Repeat for all values in a series:

for ⟨*iterator*⟩ **do**
    instructions
end for
for all ⟨*iterator*⟩ **do**
    instructions
end for

Subroutines

Procedures

procedure ⟨*name*⟩(⟨*params*⟩)
  block
end procedure

- may change the values of the parameters
- use **return** to cause immediate exit from the procedure

Functions

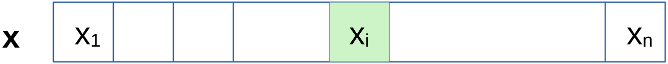
function ⟨*name*⟩(⟨*params*⟩)
  body
  return value
end function

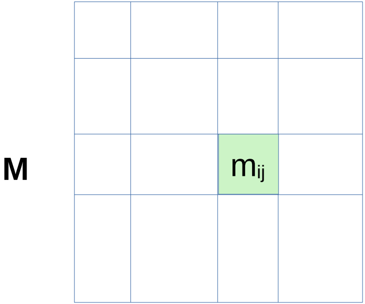
- does not change the values of the parameters
- returns a computed value

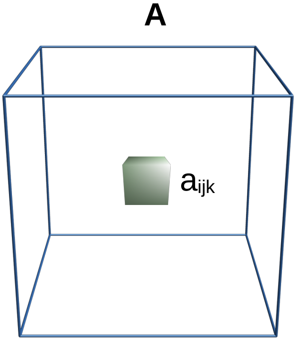
Vectors and arrays

- may contain ≥ 0 elements
- all elements have the same type (e.g. $\mathbb{N}, \mathbb{R}$)
- each element is addressble by an index - e.g. $i \in \mathbb{N}^*$ or $i, j \in \mathbb{N}^*$
- the indexing induces an order among the elements
- for the purpose of this introductory course, we stick to single element addressing

$\mathbf{x} \in \mathbb{R}^n : [x_i]; \quad \mathbf{M} \in \mathcal{M}_{m \times n}(\mathbb{R}) : [m_{ij}]; \quad \mathbf{A} \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$







Arrays - access patterns - examples

Access all $[x_i]$ sequentially:

for $i = 1, 2, \ldots, n$ **do**
  work with x_i
  $\ldots$
end for

Access all elements in odd-numbered columns:

for $j = 2k+1, k = 1, \ldots, \lfloor (n-1)/2 \rfloor$ **do**
  for $i = 1, \ldots, m$ **do**
    work with m_{ij}
    $\ldots$
  end for
end for

Sets

- bag of elements, usually of the same type
- no inherent ordering
- needs an element selection strategy to be specified
- you can use sets operations to make things clear(er)
- e.g. $A \subset \mathbb{Z}$, $|A| = n$
- if asked to detail some operation (e.g. union), then you need to describe the algorithm, not only say "$A \cup B$"

Problem 1

Find the minimum and maximum of a (a) vector, (b) matrix, and (c) set of real numbers.

Solution 1(a)

Input: $x = [x_i] \in \mathbb{R}^n$
Output: $x_{min} = \min_i(x); x_{max} = \max_i(x)$

```
x_min ← x_1; x_max ← x_1                    ▷ initial values
for i = 2, ..., n do
    if x_i > x_max then
        x_max ← x_i                         ▷ a larger value was found
    end if
    if x_i < x_min then
        x_min ← x_i                         ▷ a smaller value was found
    end if
end for
```

Solution 1(b)

Input: $X = [x_{ij}] \in \mathcal{M}_{m,n}(\mathbb{R})$
Output: $x_{min} = \min_{ij}(X); x_{max} = \max_{ij}(X)$

```
x_min ← x_11; x_max ← x_11                        ▷ initial values
for i = 1, ..., m do
    for j = 1, ..., n do
        if x_ij > x_max then
            x_max ← x_ij                          ▷ a larger value was found
        end if
        if x_ij < x_min then
            x_min ← x_ij                          ▷ a smaller value was found
        end if
    end for
end for
```

Solution 1(c)

Input: $A \subset \mathbb{R}, A \neq \emptyset$
Output: $a_{min} = \min(A); a_{max} = \max(A)$

$a_{min} \leftarrow \infty; a_{max} \leftarrow -\infty$ ▷ initial values
while $A \neq \emptyset$ **do**
  $a \leftarrow$ pick random element from A
  if $a > a_{max}$ **then**
    $a_{max} \leftarrow a$ ▷ a larger value was found
  end if
  if $a < a_{min}$ **then**
    $a_{min} \leftarrow a$ ▷ a smaller value was found
  end if
  $A \leftarrow A \setminus \{a\}$ ▷ remove the element from A
end while

Problem 2

Given a vector of real numbers, sort its elements in increasing order.

Solution 2 (Bubble sort)

Input: $x = [x_i] \in \mathbb{R}^n$
Output: sorted x

repeat
  swapped $\leftarrow$ *False*
  for $i = 1, \ldots, n-1$ **do**
    if $x_i > x_{i+1}$ **then**
      $t \leftarrow x_i$ ▷ need to swap x_i and x_{i+1}
      $x_i \leftarrow x_{i+1}$
      $x_{i+1} \leftarrow t$
      swapped $\leftarrow$ *True*
    end if
  end for
until not swapped

Problem 3

Given a real-valued vector x, compute the (sample-based) estimates of the mean and standard deviation.

Solution 3 - first version

Use $\hat{\mu} = \frac{1}{n} \sum_i x_i; \quad \hat{\sigma}^2 = \frac{1}{n-1} \sum_i (x_i - \hat{\mu})^2$

Input: $x = [x_i] \in \mathbb{R}^n$

Output: m - the mean; σ - the standard deviation

$m \leftarrow 0$
for $i = 1, \ldots, n$ **do**
  $m \leftarrow m + x_i$
end for
$m \leftarrow \frac{m}{n}$
$s \leftarrow 0$
for $i = 1, \ldots, n$ **do**
  $s \leftarrow s + (x_i - m)^2$
end for
$\sigma \leftarrow \sqrt{\frac{s}{n-1}}$

Discussion

- why not use updates like $m \leftarrow m + \frac{x_i}{n}$?
- what happens if $n = 1$?
- what about underflow and precision of the result? how can we improve the robustness?
- can we make it faster - e.g. by passing only once through data?

Solution 3 - second version

It can be show (prove it!) that

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_i (x_i - \mu)^2 = \frac{1}{n-1}\sum_i x_i^2 + \frac{1}{n(n-1)}\left(\sum_i x_i\right)^2$$

Input: $x = [x_i] \in \mathbb{R}^n$
Output: m - the mean; σ - the standard deviation

$s_1 \leftarrow 0; s_2 \leftarrow 0$
for $i = 1, \ldots, n$ **do**
  $s_1 \leftarrow s_1 + x_i$
  $s_2 \leftarrow s_2 + x_i^2$
end for
$m \leftarrow \frac{s_1}{n}$
$\sigma \leftarrow \sqrt{\frac{s_2}{n-1} + \frac{s_1^2}{n(n-1)}}$

Questions?