Real-world Algorithms

...a lightning introduction to algorithms

Area-Circumference Problem

Requirements:

Design an algorithm to find and report the area and circumference of a circle whose radius the user gives.

Example interaction

```
Welcome to circle computer…
Please enter the radius: 5
The area of a circle of radius 5 is 78.55
and its circumference is 31.42
```

Area-Circumference Problem

To find area & circumference of circle…

First say “Welcome to circle computer…” and then ask the user for the radius of their circle and get the radius value the user gives in response. Next, compute the corresponding area as pi times the radius squared, and then compute the circumference as two times pi times the radius. Finally, tell the user that the area of a circle of the radius they gave is the area value you computed, and the circumference is the circumference value you computed.

Area-Circumference Problem

To find area & circumference of circle…

First say "Welcome to circle computer…" and then ask the user for the radius of their circle and get the radius value the user gives in response. Next, compute the corresponding area as pi times the radius squared, and then compute the circumference as two times pi times the radius. Finally, tell the user that the area of a circle of the radius they gave is the area value you computed, and the circumference is the circumference value you computed.

Area-Circumference Problem

To find area & circumference of circle…

First say "Welcome to circle computer…"

and then ask the user for the radius of their circle and get the radius value the user gives in response

Next, compute the corresponding area as pi times the radius squared,

and then compute the circumference as two times pi times the radius.

Finally, tell the user that the area of a circle of the radius they gave is the area value you computed, and the circumference is the circumference value you computed.

Area-Circumference Problem

To find area & circumference of circle…

1. **Say** "Welcome to circle computer…"
2. **Ask** the user for the radius of their circle and **get** the *radius* value the user gives in response
3. **Compute** the corresponding *area* as *pi* times the *radius* squared,
4. **Compute** the *circumference* as two times *pi* times the *radius*.
5. **Tell** the user that the area of a circle of the *radius* they gave is the *area* value you computed, and the circumference is the *circumference* value you computed.

Area-Circumference Problem

To find area & circumference of circle…

1. Print welcome message
2. Ask for & get radius from user
3. Compute area as pi.radius.radius
4. Compute circumference as 2.pi.radius
5. Report area, circumference & radius

Identify preconditions
& ensure they are satisfied.

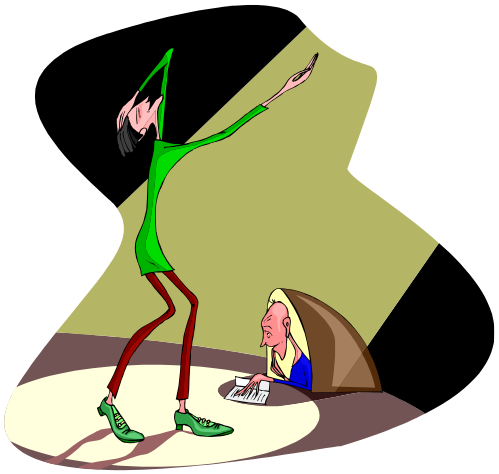
- Step 3 requires radius value, obtained from 2. Impossible to do 3 before 2.
- Step 5 requires 3 & 4, so must come last!
  - Steps 3 & 4 may be done in either order
- Need to know the value of pi is 3.142

Once we are sure that this is correct, move on to solve any non-trivial sub-problems.

Area-Circumference Problem

Solve…

2. **Ask for & get radius from user**
    1. Ask (prompt) the user to enter radius
    2. Get radius value from user



Failing to prompt user leaves them wondering what they are supposed to enter

E.g., please tell me the radius of your circle…
Then listen for and note down the response.

Area-Circumference Problem

Solve…

5. Report area, circumference & radius

1. Print msg “The area of a circle with radius ”
2. Print radius value
3. Print msg “ is ”
4. Print area value and move to next line
5. Print msg “ and its circumference is ”
6. Print circumference value
7. Print blank line

Take care with spaces, number formats & line breaks!

Exam average problem

Requirements:

Given a set of already graded exam papers,
compute and report the average grade.

Algorithm

1. Print welcome message
2. Given the set of exam papers
   find the number of papers and
   the sum of all the grades on the papers
3. Compute average grade
   as sum of all the grades / number of papers
4. Report the average grade
5. Print "all done" message

Exam average problem

- What preconditions inherent in last step?
- After sketching out an initial algorithm, check it before progressing
- Can you see any problems?
- Notice problem of no papers,
  - divide by zero error!
  - run-time error (three types of error, syntax, logical & runtime)
  - Rewrite the algorithm to avoid this error
  - How would you express this solution in English?

- Using “if”
  - Two forms:
    - if … then …
    - if … then … else …

Exam average problem

Revised algorithm

Algorithm

1. Print welcome message
2. Given the set of exam papers
   find the number of papers and
   the sum of all the grades on the papers
3. **if** number of papers is zero **then**

3T Print msg "no grades entered"

   else

3F.1 Compute average grade
   as sum of all the grades / number of papers

3F.2 Report the average grade

4. Print "all done" message

Exam average problem

Solve…

2. **Given the set of exam papers**
find the number of papers and
the sum of all the grades on the papers

1. Ask user for & get the number of papers
2. For each paper read the grade from the paper and add it to the sum of grades so far.
3. Sum of all the grades is now sum of grades so far

Do you see any problems?

Is there an alternative to asking the user how many papers there are?

Exam average problem

Solve… (Alternative)

2. **Given the set of exam papers**
find the number of papers and
the sum of all the grades on the papers

1. Set count of papers so far to zero, then for each paper add one to the count of papers so far.
2. Set sum of grades so far to zero, and then for each paper read the grade from the paper and add it to the sum of grades so far.
3. Number of papers is now count of papers so far
4. Sum of all grades is now sum of grades so far

Can we do Step 1 and 2 together?

Exam average problem

Solve… (Yet another alternative)

2. **Given the set of exam papers**
find the number of papers and
the sum of all the grades on the papers

1. Set count of papers so far to zero
2. Set sum of grades so far to zero
3. For each paper, read the grade from the paper, add it to the sum of grades so far and add one to the count of papers so far
4. Number of papers is now count of papers so far
5. Sum of all grades is now sum of grades so far

Step 3 (sub-step 2.3) is looking complicated: rewrite

Exam average problem

Solve… (& yet another alternative)

2. Given the set of exam papers find the number of papers and the sum of all the grades on the papers

1. Set count of papers so far to zero
2. Set sum of grades so far to zero
3. For each paper

3.1 read the grade from the paper,
3.2 add grade to the sum of grades so far
3.3 add one to the count of papers so far

4. Number of papers is now count of papers so far
5. Sum of all grades is now sum of grades so far

Exam average problem

- In English, what other ways might you phrase step 3

(For each paper …)

  - while there are more papers do …,
  - repeat … until there are no more papers,
  - do … while there are more papers
  - for all papers do …
  - for every paper do …

- Any other problems?

  - Customers now decide they want the program to check for valid grades
  - Give an error message and reject any grade outside of the range 0-100.
  - This requires “if” too. How would you say in English?

Sentence forms - control

- Sequence
  - "Do this and then do that and then do the other and …"
  - Put each step on a different line
- Decision/alternation
  - "**if** this condition is true **then** do this **else** do that"
  - "**if** this condition is true **then** do this"
  - Indent the actions that will be done in case the condition is true or false
- Repetition
  - "**for** each/every/all **do** this"
  - "**repeat** this **until** condition is true"
  - "**while** this condition is true **do** this"
  - "**do** this **while** condition holds"
  - Indent the action that will be repeated.

all computable problems can be solved using only sequence, decision & repetition.

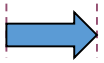
Types & Layout *(of algorithm steps)*

Sequence

1. step
2. step
3. step
4. step

Decision

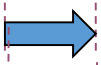
n. **if** condition **then**
nT step
else
nF step



indent

Repetition

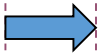
n. **while** condition **do**
step



indent

n. **for** so many times **do**
step

n. **do**
step
while condition



indent

Sentence forms - data

- Things that will be done (actions) fall into one of only three categories
- Input
  - “get value from user”
- Computation
  - “compute this as some function of that”
- Output
  - “print this message”
  - “print/report this value”

More Examples on Real World Algorithms

Example 2

Write the pseudocode of an algorithm for finding the highest number from a sequence of n numbers:

Pick the first number and call it "the highest so far".

For each number in the sequence

    If it is higher than the "highest so far"

        Discard "the highest so far".

        Call this number "the highest so far".

The number called "the highest so far" is the highest number in the sequence.

What may go wrong?

Example 3

Suppose you have a random sequence of black and white marbles and want to rearrange it so that the black and white marbles are grouped together.

Continued

Example 3

Suppose you have a random sequence of black and white marbles and want to rearrange it so that the black and white marbles are grouped together.

Repeat until sorted

Locate the first black marble that is preceded by a white marble, and switch them.

Continued

Example 3

Repeat until sorted

Locate the first black marble that is preceded by a white marble, and switch them.

What does the algorithm do with this sequence:

Spell out the steps until the algorithm stops (trace the algorithm).

Continued

Example 3

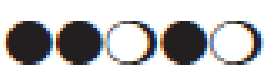
The first black marble that is preceded by a white one is marked in blue:

Switching the two yields

The next black marble to be switched is

yielding

The next steps are



Example 4

Suppose you have a random sequence of colored marbles.

Consider this pseudocode:

Repeat until sorted
    Locate the first marble that is preceded by
    a marble of a different color, and switch them.

Is this an algorithm?

Answer: The sequence doesn't terminate. Consider the input:
The first two marbles keep getting switched.