

FLUENCY6

with information technology

SKILLS, CONCEPTS, & CAPABILITIES



LAWRENCE SNYDER

Chapter 10

Algorithmic Thinking

Learning Objectives

- Explain similarities and differences among algorithms, programs, and heuristic solutions
- List the five essential properties of an algorithm
- Use the Intersect Alphabetized List algorithm to do the following:
 - Follow the flow of the instruction execution
 - Follow an analysis to pinpoint assumptions
- Demonstrate algorithmic thinking by being able to do the following:
 - Explain the importance of alphabetical order on the solution
 - Explain the importance of the barrier abstraction for correctness

The Letter Algorithm

- Bauby was paralyzed and could communicate only by blinking one eyelid
- An assistant would say or point to a letter and Bauby would indicate if it was the right one
- Point to letters until the correct one is reached
- Repeat to spell words and sentences

The Letter Algorithm

- This process is an algorithm: A precise, systematic method for producing a specified result
- We invent algorithms all the time
- An algorithm need not use numbers
- The agent running an algorithm may be a human being, rather than a computer
- There are better and poorer versions of this algorithm

The Letter Algorithm

- Making the process faster
 - _ Completion: The assistant can guess a word before it is all spelled out
 - _ Ask the letters in frequency order, and work from the most-frequently-used letter downward

Programs are Algorithms

- **programs:** algorithms that have been specialized to a specific set of conditions and assumptions
 - usually written in a specific programming language
- usually the words program and algorithm are used interchangeably

Algorithms we have learned

- Placeholder technique
 - protect “good” letter sequences by replacing them with a placeholder
 - edit the “bad” letter sequences
 - restore the “good” letter sequences
- Binary to Decimal Conversion
 - if there is a 1, write down the place value for its position in decimal
 - add up those place values
- Binary Addition
 - add as in decimal but limit digit positions to two

Algorithms vs. Heuristic Processes

- not all processes given in the book are algorithms
- the process to find information on the web using a search engine was not an algorithm
 - not systematic
 - not guaranteed to find it (process could fail)
 - called a **heuristic process**: helpful procedure for finding a result

Algorithm Properties

- An algorithm *must* have five properties:
 1. Input specified
 2. Output specified
 3. Definiteness
 4. Effectiveness
 5. Finiteness

1. Input Specified

- The **input** is the data to be transformed during the computation to produce the output
- What data do you need to begin to get the result you want?
- Input precision requires that you know what kind of data, how much and what form the data should be

2. Output Specified

- The output is the data resulting from the computation (your intended result)
- Frequently the name of the algorithm contains the output:
 - “Algorithm to compute batting average”
- Output precision also requires that you know what kind of data, how much and what form the output should be (or even if there will **be** any output at all!)

3. Definiteness

- Algorithms must specify every step and the order the steps must be taken in the process
- Definiteness means specifying the sequence of operations for turning input into output
- Details of each step must be spelled out (including how to handle errors)

4. Effectiveness

- For an algorithm to be effective, each of its steps must be doable
- The agent must be able to perform each step without any outside help or extraordinary powers

5. Finiteness

- The algorithm must stop, eventually!
- Stopping may mean that you get the expected output OR you get a response that no solution is possible
- Finiteness is not usually an issue for non-computer algorithms
- Computer algorithms often repeat instructions with different data and finiteness may be a problem

Query Evaluation

1. Parse the query.
 2. Convert words into wordIDs.
 3. Seek to the start of the doclist in the short barrel for every word.
 4. Scan through the doclists until there is a document that matches all the search terms.
 5. Compute the rank of that document for the query.
 6. If we are in the short barrels and at the end of any doclist, seek to the start of the doclist in the full barrel for every word and go to step 4.
 7. If we are not at the end of any doclist go to step 4.
- Sort the documents that have matched by rank and return the top k.

Figure 4. Google Query Evaluation

Figure 10.4 Original query evaluation algorithm developed by Brin and Page to find the hits for a Google search; from their original paper, *The Anatomy of a Large-Scale Hypertextual Web Search Engine* (infolab.stanford.edu/~backrub/google.html).

Query Evaluation

- Makes an ordered list of the pages after a search query
- Not written in a programming language, instead just everyday English
 - Does use “tech speak”
- Writing in English instead of a programming language allows them to omit details
- They are writing for people, not computers

Algorithm Fact #1

1. Algorithms can be specified at different levels of detail
 - Algorithms use functions to simplify the algorithmic description
 - These functions (such as *scan*) may have their own algorithms associated with them

Algorithm Fact #2

2. Algorithms always build on functionality previously defined and known to the user
 - Assume the use familiar functions and algorithms
 - For example, “scan through” would use the Intersecting Alphabetized Lists (IAL) from Chapter 5

Algorithm Fact #3

3. Different algorithms can solve the same problem differently, and the different solutions can take different amounts of time (or space)

Intersecting Lists

Here is the Intersecting Alphabetized Lists (IAL) algorithm from Chapter 5

1. Put a marker at start of each list
2. If all markers show the same URL, save it
3. Advance the marker(s) for the alphabetically-earliest URL
4. Repeat steps 2-3 until some marker reaches the end

No Alphabetized Lists

1. Arrange lists left to right on the page
2. Put a marker at the start of each list
3. If all markers show same URL, save it
4. Starting from the right, find each list whose marker is at the end, and move the marker to the front
5. When you first find a marker not at the end, advance it and go to step 3.
6. Stop when all markers return to start

How Not To Match

- IAL and NAL are different algorithms
- IAL and NAL accomplish the same thing
- NAL is much slower
- For 5 lists of 10 URLs:
 - IAL: $10+10+10+10+10 = 50$ steps
 - NAL: $10 \times 10 \times 10 \times 10 \times 10 = 100,000$ steps

How Do We Know it Works?

- Algorithm solution is clear and simple and efficient
- Then, how do we know it works?
- If there is no loop, the program runs, gets to an end, and we can check the result
- What if there is a loop?
 - Programs with loops cannot be absolutely verified...there are too many possible cases

Then, what?

- *The way to know that an algorithm works is to know why it works...*
- Strategy for knowing why it works:
 - Find one or more properties that ensure the algorithm works
 - Explain, using the program, why they make it work.

Why IAL Works

- If a URL appears in all lists, it forms a barrier
- Pointers that haven't reached the barrier move sooner than ones that have
- When they all reach the barrier, the URL is recorded
- None can pass the next barrier until they all reach it
- Etc.

Correctness

- anyone who creates an algorithm needs to know why it works
 - finding the algorithm's correctness-preserving properties and explaining why they do the job

Summary

- We use algorithms daily, and we continually create them as we instruct other people in how to do something
- Everyday algorithms can be sometimes be unclear because natural language is imprecise
- Algorithms have five fundamental properties

Summary

- Algorithms can be given at different levels of detail depending on the abilities of the agent
- Problems can be solved by different algorithms in different ways
- Algorithms always finish—either they give the answer, or say no answer is possible—and they are evaluated on their use of resources such as space and time

Summary

- *Intersect Alphabetized Lists* algorithm is used in Web search, and is preferred over other solutions
- Two properties of the *Intersect Alphabetized Lists* algorithm tell us why it works: the Alpha-Ordering and Barrier properties