TSSM™

Creating VCE Success

Software Development Teach Yourself Series

Topic 7: Sorting and Searching Algorithms Units 3&4

A: Level 14, 474 Flinders Street Melbourne VIC 3000
T: 1300 134 518 **W:** tssm.com.au **E:** info@tssm.com.au

Contents

CASE STUDY

All Teach Yourself Series in this package will refer to the following case study.

Tariq Mulner is the manager of a school canteen. He manages how many lunches are going to be prepared each day. It is difficult to tell how many lunches will be sold, so he would like a software solution that students can use to order lunches. This application would provide him with a complete list of orders.
Most of the students have smart phones, so Tariq is suggesting the solution is a phone app that can read in the lunch order, and send it to his device so he can print out the order list.

Photo by Tirachard Kumtanom from Pexels used with permission

Algorithms

An algorithm is a solution to a logic problem. In the Study Design we use Pseudocode to write algorithms.

The Features of Pseudocode

- Every algorithm must begin and end with START and STOP or BEGIN and END.
- When assigning data to a variable use ←. Example: IntNumber ← 5
- Show each calculation required. Example: IntAnswer ← IntNumber1 + IntNumber2
- Displaying data as output Example: lblAnswer.Text ← IntAnswer or Display IntAnswer
- Decision Control Structures: Example: IF condition THEN Action1 ELSE Action2 ENDIF
- Case Control Structures:
  Example: Case where condition, CASE1 Action1 CASE2 Action2 END CASE
- Repetition Control Structures: Example: FOR counter ← first value to last value END FOR
- Repetition Control Structures: Example: REPEAT actions UNTIL condition
- Repetition Control Structures: Example: WHILE condition DO actions END WHILE
- Use indentation and spaces to illustrate how control structures are incorporated

Here is an example of a pre-test loop allowing three attempts at entering a username and password.

```
START
    strUsername ← BillBurr123
    strPassword ← Summer1965
    Read in strUsernameEntered
    Read in strPassswordEntered
    Counter ← 0
    WHILE strUsername <> strUsernameEntered AND strPassword <> strPassswordEntered AND Counter < 3
        Display Prompt "Your Username and Password are incorrect please try again!"
        Read in strUsernameEntered
        Read in strPassswordEntered
        Counter ← Counter +1
    END WHILE
    IF Counter < 3 THEN
        IF strUsername == strUsernameEntered AND strPassword == strPassswordEntered
            Display Prompt "Your Username and Password are correct!"
        ELSE
            Display Prompt "You have exhausted your attempts. Please contact SysAdmin!"
        END IF
    END IF
END
```

Arrays

In Software Development we use Arrays to clearly demonstrate how data can be manipulated with algorithms. Array data has an index that allows us to access the data and relocate it if necessary.

Array Example: Names(6) holds 6 names.

0	1	2	3	4	5
David	Daphne	Doug	Delphi	Daniel	Dobby

Searching

In Software Development we investigate two search algorithms.

1. Linear Search
2. Binary Search

Linear Search

The simplest search algorithm is Linear. It basically starts at the very first item, check if it is the data we are looking for, if it isn't, it checks the next one. This continues until the item is found or until end of file (EOF).

Advantages

- Easy to program
- Good for use with small files

Disadvantages

- Inefficient
- Takes too long to find data in large files

We are going to write a linear search for Delphi in Names(6).

0	**1**	**2**	**3**	**4**	**5**
David	Daphne	Doug	Delphi	Daniel	Dobby

```
START
        Counter ← 0
        Found ← False
        WHILE Found = False AND Not EOF DO
                If Names (Counter) <> Delphi THEN
                        Counter ← Counter + 1
                ELSE
                        Found ← TRUE
                END IF
        END WHILE
END
```

Trace Table		
Counter	Found	Names(Counter)
0	False	David
1	False	Daphne
2	False	Doug
3	True	Delphi

The number of items in the array is 6 (n). The worst case for finding any item with linear is if the item is the last in the array. This means if Delphi was in position 5 it would take 6(n) operations or repeats through the loops to check each item.

This is written $O_{(n)}$.

Solutions to Review Questions

1. B. Linear

2. D. 2

3. D. 5

4. C. Binary

5. Create a TRACE TABLE to find out how many Hamburgers are sold each day using Binary Search.

Trace Table						

Found	**Lowest**	**Highest**	**Middle**	**Products(Lowest,0)**
				Products(Lowest,1)
					Products(Highest,0)
					Products(Highest,1)
						Products(Middle,0)
						Products(Middle,1)

False	0	6	3	Beef Pies 23	Tacos 24	Hot Dogs 8
False	0	2	1	Beef Pies 23	Egg Sandwiches 23	Hamburgers 7
True						

6. Second PASS.

0	1	2	3
Aaron	Archie	Christian	Betty