



Standard Algorithms





Standard Algorithms

Standard algorithms are sequences of code which are used regularly in different programs to solve a particular problem.

The main algorithms you need to know at Higher are:

- Input Validation (*N5 revision*)
- Find Minimum
- Find Maximum
- Count Occurrences
- Linear Search



Find Minimum

Find Minimum is used to identify the smallest value in an array

Ages	
0	15
1	12
2	7
3	16
4	12



If used on the Ages array above, Find Minimum would return the value, 7



Find Minimum

1. SET *min* TO *ages*[0]
2. FOR *counter* FROM 1 TO 4 DO
3. IF *ages*[*counter*] < *min*
4. SET *min* TO *ages*[*counter*]
5. END IF
6. END FOR
7. SEND "The lowest age is"& *min* TO DISPLAY

1.Min is initialised to match first item in array

2.Repeat for each item in array starting at item 2

3.Check if current array item is lower than Min

4.If true, set Min to match current array item



Find Maximum

Find Maximum is used to identify the largest value in an array

Ages	
0	15
1	12
2	7
3	16
4	12



If used on the Ages array above, Find Maximum would return the value, 16



Find Maximum

1. SET *max* TO *ages*[0]
2. FOR *counter* FROM 1 TO 4 DO
3. IF *ages*[*counter*] > *max*
4. SET *max* TO *ages*[*counter*]
5. END IF
6. END FOR
7. SEND "The highest age is"& *max* TO DISPLAY

1. Max is initialised to match first item in array

2. Repeat for each item in array starting at item 2

3. Check if current array item is higher than Max

4. If true, set Max to match current array item



Count Occurrences

Count Occurrences is used to identify how many times a particular value appears in an array

Names	
0	Fred
1	Betty
2	Wilma
3	Betty
4	Barney

Betty
2

If used on the Names array above for the name “Betty”, Count Occurrences would return the value, 2



Count Occurrences

1. RECEIVE *target* FROM KEYBOARD
2. SET *numFound* TO 0
3. FOR *counter* FROM 0 TO 4
4. IF *names[counter] = target*
5. SET *numFound* TO *numFound* + 1
6. END IF
7. END FOR
8. SEND "The number found is"& *numFound* TO DISPLAY

1. Ask user to enter value to count

2. Initialise *numFound* to 0

3. Repeat for each item in array

4. Check if current name matches target

5. If true, increment *numFound* by 1



Linear Search

Linear Search is used to identify whether or not an item is in a list, and which position it occupies

Names	
0	Fred
1	Betty
2	Wilma
3	Betty
4	Barney

Betty
FOUND
at position 3

If used on the Names array above for the name, “Betty”, Linear Search would return the position, 3

Linear Search



1. RECEIVE *target* FROM KEYBOARD

2. SET *found* TO FALSE

3. SET *position* TO 0

4. FOR *counter* FROM 0 TO 4

5. IF *names[counter]* = *target*

6. SET *found* TO TRUE

7. SET *position* TO *counter*

8. END IF

9. END FOR

10. IF *found* = TRUE

11. SEND "Found at position"& *position* TO DISPLAY

12. ELSE

13. SEND "Not found"

14. END IF

1. Ask user to enter value to find

2-3. Initialise found "flag" to False and position to 0

4. Repeat for each item in array

5. Check if current name matches target

6-7. If true, set found "flag" to true and position to current loop value

Linear Search



1. RECEIVE *target* FROM KEYBOARD

2. SET *found* TO FALSE

3. SET *counter* TO -1

4. REPEAT

5. SET *counter* TO *counter* +1

6. IF *names[counter]* = *target*

7. SET *found* TO TRUE

8. END IF

9. UNTIL *counter* =4 OR *found* = TRUE

10. IF *found* = TRUE

11. SEND "Found at position"& *position* TO DISPLAY

12. ELSE

13. SEND "Not found"

14. END IF

1. Ask user to enter value to find

2-3. Initialise found "flag" to False and loop counter to -1

4-5. Start conditional loop and increment loop counter

6. Check if current name matches target

7. If true, set found "flag" to true

9. Until item found or list is complete