AH Computing Science

STANDARD ALGORITHMS – **BINARY SEARCH**

Standard algorithms – Linear Search **(cfe H)**

AH

From Higher level we are familiar with the LINEAR Search Algorithm used to search for target values in a list of items.

In this algorithm, a conditional loop is used to compare each item to the Target. the loop terminates when the Target is found.

A boolean variable is set to false at the beginning, then set to true when the item is found. A counter is used to keep track of where the item was found and the algorithm stops when either the first occurrence of the item has been found and the boolean variable has been set to true or the end of the array has been reached.

Standard algorithms – Linear Search **(cfe H)**

AH

SET found to false

SET pos to 0

RECEIVE target FROM user

REPEAT 'for each index in the array

IF array(item) = target then

SET found to true

SET pos to index

LOOP until found = True or end of the array

Standard algorithms – Linear Search **(cfe H)**

AH

Linear Search is a **'Brute Force'** algorithm with potentially every item in the array being compared. This is not very efficient.

This must be done when arrays are not sorted.

Once an array has been sorted there are more efficient ways of searching the array

Standard algorithms – **binary search**

AH

Binary search is another common searching algorithm which involves splitting the list into 2 halves – therefore 'binary' search. The basic idea is

With a **sorted** list:

Repeat
Find the middle item in the list
If middle < target then Set beginning of list to middle+1
If middle > target then Set the end of list to middle -1
If middle = target then set found to TRUE
UNTIL found is TRUE or List is empty

Standard algorithms – **binary search**

AH

```
PROCEDURE binarySearch(ARRAY OF INTEGER myArray)

DECLARE found AS BOOLEAN INITIALLY false
DECLARE startPos INITIALLY 0
DECLARE endPos  INITIALLY length(myArray)
DECLARE middle AS INTEGER INITIALLY 0
DECLARE searchKey INITIALLY 0

RECEIVE searchKey FROM KEYBOARD
REPEAT
 SET middle TO (startPos + endPos) / 2
 # use integer division in case it is an odd number
 IF array[middle] < searchKey THEN
  SET startPos TO middle + 1
 ELSE
  SET endPos TO middle -1
 END IF
 IF array[middle] = searchKey THEN
  SET found TO true
  SEND "Found at position "& middle TO DISPLAY
 END IF
UNTIL found = true OR ( startPos > endPos )
 IF found = false THEN
  SEND "Not found" TO DISPLAY
 END IF

END PROCEDURE
```

Standard algorithms – Comparing Binary and Linear Search

AH

Linear

- Simple Algorithm
- Data can be unsorted
- Average number of comparisons needed **n/2**
- Slow for large lists

Binary

- Complex Algorthim
- **Data must be sorted**
- Average number of comparisons needed is **x** where **2^x>n**
- Fast for large lists

Standard algorithms – **binary search**

AH

List size	Number of comparisons
3	2 ($2^2 = 4$)
7	3 ($2^3 = 8$)
15	4 ($2^4 = 16$)
28	5 ($2^5 = 32$)
60	6 ($2^6 = 64$)
. . .	. . .
5978	16 ($2^{15} = 65536$)