****

Software Development: Data Structures

(H16Y 35)

Data Search & Sort - Introduction

Refer to the online resource links provided to carry out independent/group study on the data search and sort types listed below. Document your findings describing each search and sort types provided and upon completion, peer with another member of the class to discuss your findings.

This exercise is aimed to introduce learners to the concept of data search and sort which will be discussed in more details with worked examples and exercises in next class

Your lecturer will go around facilitating the group discussion across the teams.

**Data Search**

* Linear Search (LS)

Linear search is a very simple search algorithm. In this type of search, a sequential search is made over all items one by one. Every item is checked and if a match is found then that particular item is returned, otherwise the search continues till the end of the data collection.

Linear Search ( Array A, Value x)

Step 1: Set i to 1

Step 2: if i > n then go to step 7

Step 3: if A[i] = x then go to step 6

Step 4: Set i to i + 1

Step 5: Go to Step 2

Step 6: Print Element x Found at index i and go to step 8

Step 7: Print element not found

Step 8: Exit

procedure linear\_search (list, value)

for each item in the list

if match item == value

return the item's location

end if

end for

end procedure

* Binary Search (BS)

Search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise narrow it to the upper half. Repeatedly check until the value is found or the interval is empty.

binary\_search(A, target):

lo = 1, hi = size(A)

while lo <= hi:

mid = lo + (hi-lo)/2

if A[mid] == target:

return mid

else if A[mid] < target:

lo = mid+1

else:

hi = mid-1

// target was not found

* Differentiate between LS and BS

The major difference between linear search and binary search is that binary search takes less time to search an element from the sorted list of elements. So it is inferred that binary search method is more efficient than linear search.

Another difference between the two is that there is a prerequisite for the binary search, i.e., the elements must be **sorted** while in linear search there is no such prerequisite. Although both the searching methods use different techniques which are discussed below.

**Data Sort**

* Bubble Sort

The **bubble sort** makes multiple passes through a list. It compares adjacent items and exchanges those that are out of order. Each pass through the list places the next largest value in its proper place. In essence, each item “bubbles” up to the location where it belongs.

[Figure 1](http://interactivepython.org/runestone/static/pythonds/SortSearch/TheBubbleSort.html#fig-bubblepass) shows the first pass of a bubble sort. The shaded items are being compared to see if they are out of order. If there are n items in the list, then there are n−1n−1 pairs of items that need to be compared on the first pass. It is important to note that once the largest value in the list is part of a pair, it will continually be moved along until the pass is complete.



* Selection Sort
* Insertion Sort
* Shell Sort
* Merge Sort

Resource Link:

* <http://datastruct.hnd-computing.info/?page_id=359>
* <http://datastruct.hnd-computing.info/?page_id=382>
* <https://www.sqa.org.uk/e-learning/ArrayDS03CD/page_01.htm>