**WIA 1002 TUTORIAL 10**

**SORT AND SEARCHING**

1. Compare between linear search and binary search algorithms by searching for the numbers 45 and 54 in the following list:

**3 8 12 34 54 85 61 110**

**ANSWER:**

Linear search will search from first to last element, until the 4th element, it will return 4 to show the position of 54. However, it will search till the end then return -1 since 45 Is not in the list.

Binary search will split the list into half, middle key is 34, if the element wanted is greater than middle key, then middle key will go to the middle element of the second half of the group. Then middle element is 85, the element wanted is smaller than 85, then middle element will become the first half part, then middle element is 54, so the loop end, return the position of 54(result return 4). It same goes to 45, 45 greater than middle element, then middle element become 85, then middle element changes again, but there is no more element to compare with the middle key, so loop end, return -1. (-1 indicate search not found)

This show that linear search need 5 loops for number 54 and 8 loops for number 45, but binary search need 3(search 2 times) loops for searching both elements, so binary search is faster.

* 1. 2. Describe the technique for each sort algorithm below. Given the following list:
  2. a. Selection sort
  3. b. Insertion sort
  4. c. Bubble sort
  5. d. Merge sort

**90 8 7 56 125 237 9 1 653**

Show a trace of execution for:

ANSWER:

Selection Sort:

Find the smallest and sort it to front

Has sorted and unsorted list to arrange the element

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

idx │ 90 │ 8 │ 7 │ 56 │ 125 │ 237 │ 9 │ 1 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘



┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

[0] │ 1 │ 8 │ 7 │ 56 │ 125 │ 237 │ 9 │ 90 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘



┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

[1] │ 1 │ 7 │ 8 │ 56 │ 125 │ 237 │ 9 │ 90 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

[2] │ 1 │ 7 │ 8 │ 56 │ 125 │ 237 │ 9 │ 90 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐



[3] │ 1 │ 7 │ 8 │ 9 │ 125 │ 237 │ 56 │ 90 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘



┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

[4] │ 1 │ 7 │ 8 │ 9 │ 56 │ 237 │ 125 │ 90 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐



[5] │ 1 │ 7 │ 8 │ 9 │ 56 │ 90 │ 125 │ 237 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

Insertion Sort

Sort from the first then swap

Also divided into sorted and unsorted list, when till that element will compare it with those in the sorted list.

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

itr │ 90 │ 8 │ 7 │ 56 │ 125 │ 237 │ 9 │ 1 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐



[1] │ 8 │ 90 │ 7 │ 56 │ 125 │ 237 │ 9 │ 90 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘



┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

[2] │ 7 │ 8 │ 90 │ 56 │ 125 │ 237 │ 9 │ 90 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐



[3] │ 7 │ 8 │ 56 │ 90 │ 125 │ 237 │ 9 │ 90 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

[4] │ 7 │ 8 │ 56 │ 90 │ 125 │ 237 │ 9 │ 90 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

[5] │ 7 │ 8 │ 56 │ 90 │ 125 │ 237 │ 9 │ 90 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐



[6] │ 7 │ 8 │ 9 │ 56 │ 90 │ 125 │ 237 │ 90 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐



[7] │ 7 │ 8 │ 9 │ 56 │ 90 │ 90 │ 125 │ 237 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

[8] │ 7 │ 8 │ 9 │ 56 │ 90 │ 90 │ 125 │ 237 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

Bubble Sort:

Each round need to sort the largest element to the back, so will have many round

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

idx │ 90 │ 8 │ 7 │ 56 │ 125 │ 237 │ 9 │ 1 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘



┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

[8] │ 8 │ 7 │ 56 │ 90 │ 125 │ 9 │ 1 │ 237 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘



┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐



[7] │ 7 │ 8 │ 56 │ 90 │ 9 │ 1 │ 125 │ 237 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐



[6] │ 7 │ 8 │ 56 │ 9 │ 1 │ 90 │ 125 │ 237 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐



[5] │ 7 │ 8 │ 9 │ 1 │ 56 │ 90 │ 125 │ 237 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐



[4] │ 7 │ 8 │ 1 │ 9 │ 56 │ 90 │ 125 │ 237 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐



[3] │ 7 │ 1 │ 8 │ 9 │ 56 │ 90 │ 125 │ 237 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘



┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

[2] │ 1 │ 7 │ 8 │ 9 │ 56 │ 90 │ 125 │ 237 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

Merge Sort:

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

idx │ 90 │ 8 │ 7 │ 56 │ 125 │ 237 │ 9 │ 1 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┐ ┌─────┬─────┬─────┬─────┬─────┐

[1] │ 90 │ 8 │ 7 │ 56 │ │ 125 │ 237 │ 9 │ 1 │ 653 │

└─────┴─────┴─────┴─────┘ └─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┐ ┌─────┬─────┐ ┌─────┬─────┬─────┬─────┬─────┐

[2] │ 90 │ 8 │ │ 7 │ 56 │ │ 125 │ 237 │ 9 │ 1 │ 653 │

└─────┴─────┘ └─────┴─────┘ └─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┐ ┌─────┬─────┐ ┌─────┬─────┬─────┐

[3] │ 7 │ 8 │ 56 │ 90 │ │ 125 │ 237 │ │ 9 │ 1 │ 653 │

└─────┴─────┴─────┴─────┘ └─────┴─────┘ └─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┐ ┌─────┬─────┐ ┌─────┬─────┬─────┐

[4] │ 7 │ 8 │ 56 │ 90 │ │ 125 │ 237 │ │ 1 │ 9 │ 653 │

└─────┴─────┴─────┴─────┘ └─────┴─────┘ └─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┐ ┌─────┬─────┬─────┬─────┬─────┐

[5] │ 7 │ 8 │ 56 │ 90 │ │ 1 │ 9 │ 125 │ 237 │ 653 │

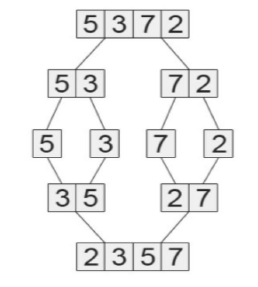
└─────┴─────┴─────┴─────┘ └─────┴─────┴─────┴─────┴─────┘

┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐

[6] │ 1 │ 7 │ 8 │ 9 │ 56 │ 90 │ 125 │ 237 │ 653 │

└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘

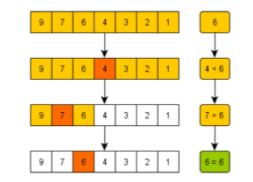
3. Which type of sort algorithm is this?



ANSWER:

Merge Sort.

4. Which type of search algorithm is this?



**Answer:**

Binary Search.

5. Which type of search algorithm is this?



**ANSWER:**

Linear Search.