MAT2440, Classwork27, Spring2025

ID:_____ Name:_____

1. Algorithms of **Searching Problems.**

Problem: Locate an element x in a list of distinct elements $a_1, a_2, \cdots, a_n$, or determine that it is **not** in the list. The **solution** to this search problem is that **location** of the term in the list that equals x (that is, __i__ is the solution if $x = a_i$) and is __0__ if x is not in the list.

Algorithms: (1) **The Linear Search** or **Sequential Search**. (2) **The Binary Search**.

2. An example of searching problem: Given a list $\{-1, 2, 5, 6, 3, -5\}$.

We have $a_1 =$__-1__, $a_2 =$__2__, $a_3 =$__5__, $a_4 =$__6__, $a_5 =$__3__, and $a_6 =$__-5__.

(1) To find the location of "3": the algorithm will return __5__. ("3" is the __5th__ element)
(2) To find the location of "1": the algorithm will return __0__. ("1" is __NOT__ in the list)

3. Searching algorithm: Algorithm and Pseudocode of the **Linear Search**.

Algorithm:

(1) Comparing x and a_1, **if** True ($x =$__=__a_1) **then** location=__1__;
if False ($x =$__≠__a_1) **then** location=__0__. continue ....
(2) Comparing x and a_2, **if** True ($x =$__=__a_2) **then** location=__2__;
if False ($x =$__≠__a_2) **then** location=__0__. continue
(3) Continue **until** $x = a_i$, **if** True ($x =$__=__a_i) **then** location=__i__.
(4) If no matching is found, then return location=__0__. ← x is NOT in the list

Pseudocode:

procedure *linear_search*(x: integer, $a_1, a_2, \cdots, a_n$: distinct integers)
$n :=$ the __length__ of $\{a_i\}$
$i :=$ __1__
while ($i \leq$ __n__ and $x \neq a_i$) → It will not stop until finds the x. or $i > n$.
    $i := i + 1$
if $i \leq$__n__ **then** *location* :=__i__
    else *location* :=__0__
return __location__ { *location* is the subscript of the term that equals x, or 0 if x is not found.}

4. An example of linear search: Let the sequence be $\{3, 1, 5, 6, 4\}$.

(a) Search for 6.

Initialization: $i =$ 1 and $n =$ 5. $x = 6$ → $i = i + 1$

While-loop:

Round	$i \le n$ and $x \neq a_i$ (T or F?)	i
$i=1$ 1	$1 \le 5 \wedge 6 \neq 3$ (T)	2
$i=2$ 2	$2 \le 5 \wedge 6 \neq 1$ (T)	3
$i=3$ 3	$3 \le 5 \wedge 6 \neq 5$ (T)	4
$i=4$ 4	$4 \le 5 \wedge 6 \neq 6$ (F)	stop

$(i \le n)$ is T implies *location* = 4. ← which is the "i" in current step.

$4 \le 5$

Return: location = 4

(b) Search for 10. $x = 10$ $\{3, 1, 5, 6, 4\}$

Initialization: $i =$ 1 and $n =$ 5. $i = i + 1$

Round	$i \le n$ and $x \neq a_i$ (T or F?)	i
$i=1$ 1	$1 \le 5 \wedge 10 \neq 3$ (T)	2
$i=2$ 2	$2 \le 5 \wedge 10 \neq 1$ (T)	3
$i=3$ 3	$3 \le 5 \wedge 10 \neq 5$ (T)	4
$i=4$ 4	$4 \le 5 \wedge 10 \neq 6$ (T)	5
$i=5$ 5	$5 \le 5 \wedge 10 \neq 4$ (T)	6
$i=6$ 6	$6 \le 5$ (F)	stop

$(i \le n)$ is False implies *location* = 0.

$6 \le 5$

Return: location = 0 ⇒ 10 is NOT in this list.