

Linear Search

ALGORITHM 2 The Linear Search Algorithm.

procedure *linear search*(x : integer, a_1, a_2, \dots, a_n : distinct integers)

$i := 1$

while ($i \leq n$ and $x \neq a_i$)

$i := i + 1$

if $i \leq n$ **then** $location := i$

else $location := 0$

return $location$ { $location$ is the subscript of the term that equals x , or is 0 if x is not found}

- What is worst case time complexity of this implementation of linear search? (Hint: it is not n .)

Big-O Discussion

For time complexity, we only care about large input sizes, and we only care about the scaling, not the detailed function.

How does big-O notation capture these two desiderata?

Big-O Practice

1. Prove that $7x + 1$ is $O(x^2)$. (NOTE: not $O(x)$).
2. Prove that $10x^2$ is not $O(x)$. (Use proof by contradiction.)