Common Search Problems:
- Is an item contained in a list?
- Where is the item in the list?
- Max/Min item
- Items in a range
- Items w/ specific type
- How many times a value appears in a list

What do we mean by performance?

Interpretation #1: Seconds to compute
\[=\] depends on hardware

Interpretation #2: Steps the algorithm takes
\[=\] hardware independent
\[=\] theoretical measure

EX Counting steps for linear search

\[L = [23, 77, -34, 95, -99, -4]\]

Alg #1

\[
\text{for } i \text{ in range(len}(L)):\n\quad \text{if } L[i] = x:\n\]

Alg #2

\[
\text{found} = \text{False}\n\text{for } i \text{ in range(len}(L)):\n\]

\[=\text{work to know the worst case}\]
\[=\text{as a function of the size of the input}\]
for i in range(len(L)):
    if L[i] == \x:
        return True
return False

found = True
for i in range(len(L)):
    if L[i] == \x:
        found = False
return found

How many steps does each alg take to find the following values?

Alg #1:
99: 5
23: 1
80: 6

Alg #2:
99: 6
23: 6
80: 6

In general:

Alg #1

\text{Ave \#steps} = \frac{N+1}{2}
\text{Min \#steps} = 1
\text{Worst case} = N

Alg #2

\text{Ave \#steps} = N
\text{Min \#steps} = N
\text{Max \#steps} = N
\text{(Worst case)} = N

We are interested in the worst-case performance
\rightarrow \text{Upper bound on running time.}
"Big-oh" notation we use to describe worst-case running time.

Linear search is $O(N)$, where $N$ is the length of the list. Because in the worst-case we need to check every element in the list.