Whiteboard

Friday, April 3, 2020 10:09 AM

Linear Search

checks each item sequentially

items can be in any order

$O(N)$

Binary Search

checks items at midpoints & eliminates halves

list must be sorted

$O(\log N)$

Rules of big-Oh

#1 Consecutive statements add:

```
x = 1          1 step
y = x + 2      2 step
return y       1 step
```

total: 4 steps

$\Rightarrow O(1)$ (constant time algorithm)

#2 for & while loop

statements inside the loop are multiplied by # iterations

```
for i in range(N):      } N steps
    print(i)            } => O(N)
```

#3 If / Else statements

→ runtime is the larger #steps for all cases

```
if x == True:          ] 1 step
    print(x)
else:
    for i in range(N):  ] N steps
        print(i)
```

$\Rightarrow O(N)$

#4 Nested for loops

```
for i in range(N):
    for j in range(N):      ] N
        print(i, j)
```

N · N Steps

$\Rightarrow O(N^2)$

NOTE: After computing steps, remove multiplier & additive constants:

EX $3N + 1 \rightarrow O(N)$

EX $3 \cdot 7 + 1 \rightarrow O(1)$

→ Only term that grows with size of input matters (usually N in our examples)

for j in range(N): $\lfloor N \rfloor$ steps
    print(i, j)
$\Rightarrow O(N^2)$

NOTE: We analyze from "inside" out

#5 Halving the data each iteration
$\rightarrow O(\log N)$