Big O notation

Announcements

- Lab 7 due tomorrow at midnight
  - Ninja session tonight, 7-9pm
- Lab 8 (searching) will be available Sunday

Today

- Review linear search vs. binary search
- Big O notation - worst case asymptotic run time
  - *O(1)*, constant
  - *O(log n)*, logarithmic
  - *O(n)*, linear
  - *O(n log n)*, linearithmic
  - *O(n^2)*, quadratic

Final analysis

- Binary search: *O(log n)*, Linear search: *O(n)*
- Binary search is faster, but only works for sorted lists.
- Linear search is easier to implement and thus less likely to contain a bug. It works for any list. For small lists, the difference in run time isn't noticeable.

Analysis of algorithms

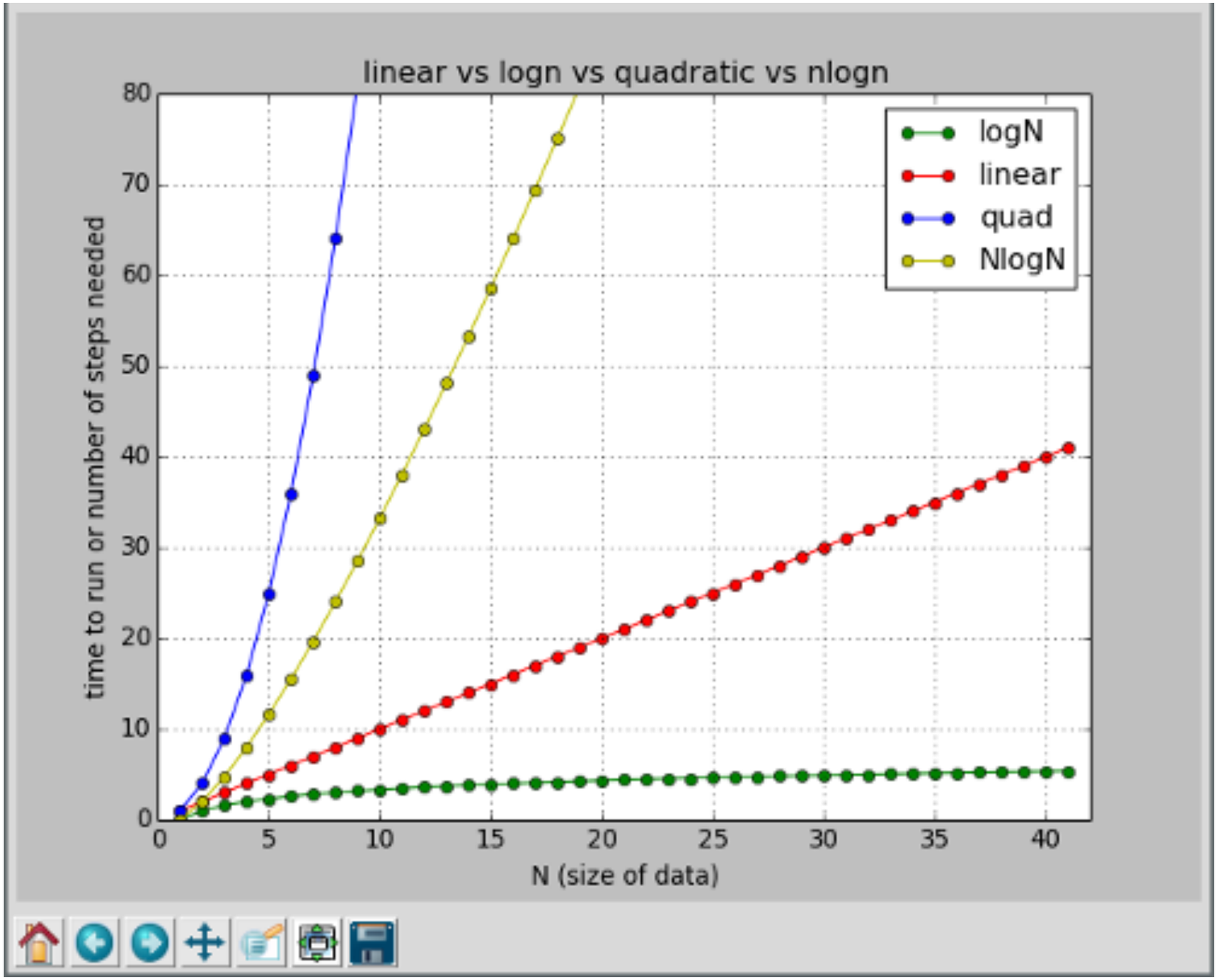
- *O(1)*: **constant** - stays the same as input size increases

- *O(log n)*: **logarithmic** - proportional to the logarithm of the input size; it increases by a constant amount each time input size doubles

- *O(n)*: **linear** - proportional to the input size; it doubles when the input size doubles

- *O(n log n)*: **linearithmic** - proportional to the input size times the logarithm of the input size; it more than doubles when input size doubles (but doesn't triple)

- $O(n^2)$: **quadratic** - proportional to the square of the input size; it quadruples when the input size doubles


linear vs logn vs quadratic vs nlogn
logN
linear
quad
NlogN
time to run or number of steps needed
N (size of data)
0
10
20
30
40
50
60
70
80
0
5
10
15
20
25
30
35
40

Examples

- *O(1)*: indexing
- *O(log n)*: binary search
- *O(n)*: linear search
- *O(n log n)*: mergesort
- $O(n^2)$: selection sort, insertion sort, bubble sort

You're hosting a party

- Number of guests is input size
- *O(1)*: cleaning your place
- $O(n^2)$: introducing every pair of guests
- *O(log n)*: slicing the bread
- *O(n)*: serving the bread

Practice classifying algorithms

- https://www.cs.swarthmore.edu/~mauskop/cs21/s17/practice/9F.html

Have a nice weekend!