15-112
Lecture 2

Sets, Dictionaries, and Efficiency

Instructor: Pat Virtue

Tuesday Logistics

www.cs.cmu.edu/~112/gallery.html

CMU 15-112, Fall 2023

Fundamentals of Programming and Computer Science
Carnegie Mellon University

W 15-112 S23 Term Project Lightning Rou...

Click here for the Term Project Gallery!

Announcements

Term Project

- Ideation Meetings
- Special Topic Sessions

Hack112

- Sat-Sun Nov 4-5
- Just for us!

HW8

- Little Alchemy!

Poll 1

Would you attend a TP ideation meeting with a TA?

- Definitely yes
- Probably yes
- Probably not
- Definitely not

Thursday Logistics

Warm-up as you walk in

Dancing is optional :-p

https://www.youtube.com/watch?v=LriMvv9qDrk

Announcements

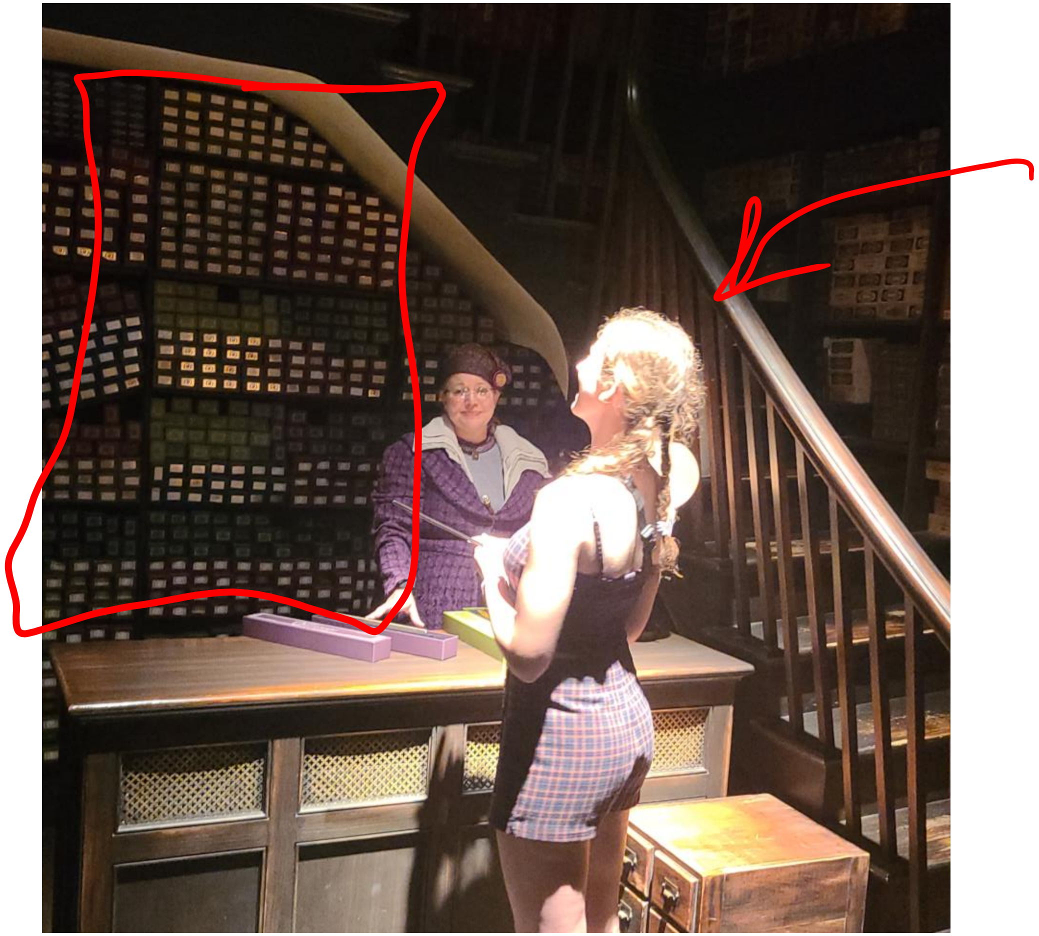
Term Project

- Preview

Sets

Debate

Which are better sets or lists?



Lists vs Sets

Property	Lists	Sets
Indexing	✓	✗
Iterate over	✓	✓
Mutable	✓	✓
Mutable elements	✓	✗
Get length	✓	✓
Check contents	✓	✓
store ~~Sort~~ different types	✓	✓
store ~~Sort~~ repeated elements	✓	✗
Stored in order added	✓	✗

“I feel the need, the need for speed”

-- Top Gun

Poll 2

Which of the following may need to visit all N elements in the list `data`, assuming N = len(data)? Select ALL that apply.

```
✓ A. for x in data:
         print(x)
✓ B. for i in range(len(data)):
         x = data[i]
         print(x)
✓ C. if x in data:
         print("Found it")
✗ D. x = data[-1]
✓ E. x = max(data)
  F. None of the above
```

Poll 3

Which of may need to visit all N elements in the set `data`, assuming N = len(data)? Select ALL that apply.

✓ A. `for x in data:`
        `print(x)`

~~B. `for i in range(len(data)):`~~
        ~~`x = data[i]`~~
        ~~`print(x)`~~

[✓ C. `if x in data:`
        `print(“Found it”)`]

~~D. `x = data[-1]`~~

✓ E. `x = max(data)`

F. None of the above

Discussion

Brainstorm: How can I make finding a specific student’s exam more efficient?

Set lookup is *way* faster

Hashing

Discussion

Brainstorm: How can I make finding a specific student’s exam more efficient?

How do sets work? In-class exercise

Hashtables

Simple example

```
def myHash(s):
    val = 0
    for c in s:
        val += ord(c)
    return val

s = "cat"

numBuckets = 10
hVal = myHash(s)
bucketIndex = hVal % numBuckets

print(s, hVal, bucketIndex)
```

How do sets work?

Hashtables

Simple example

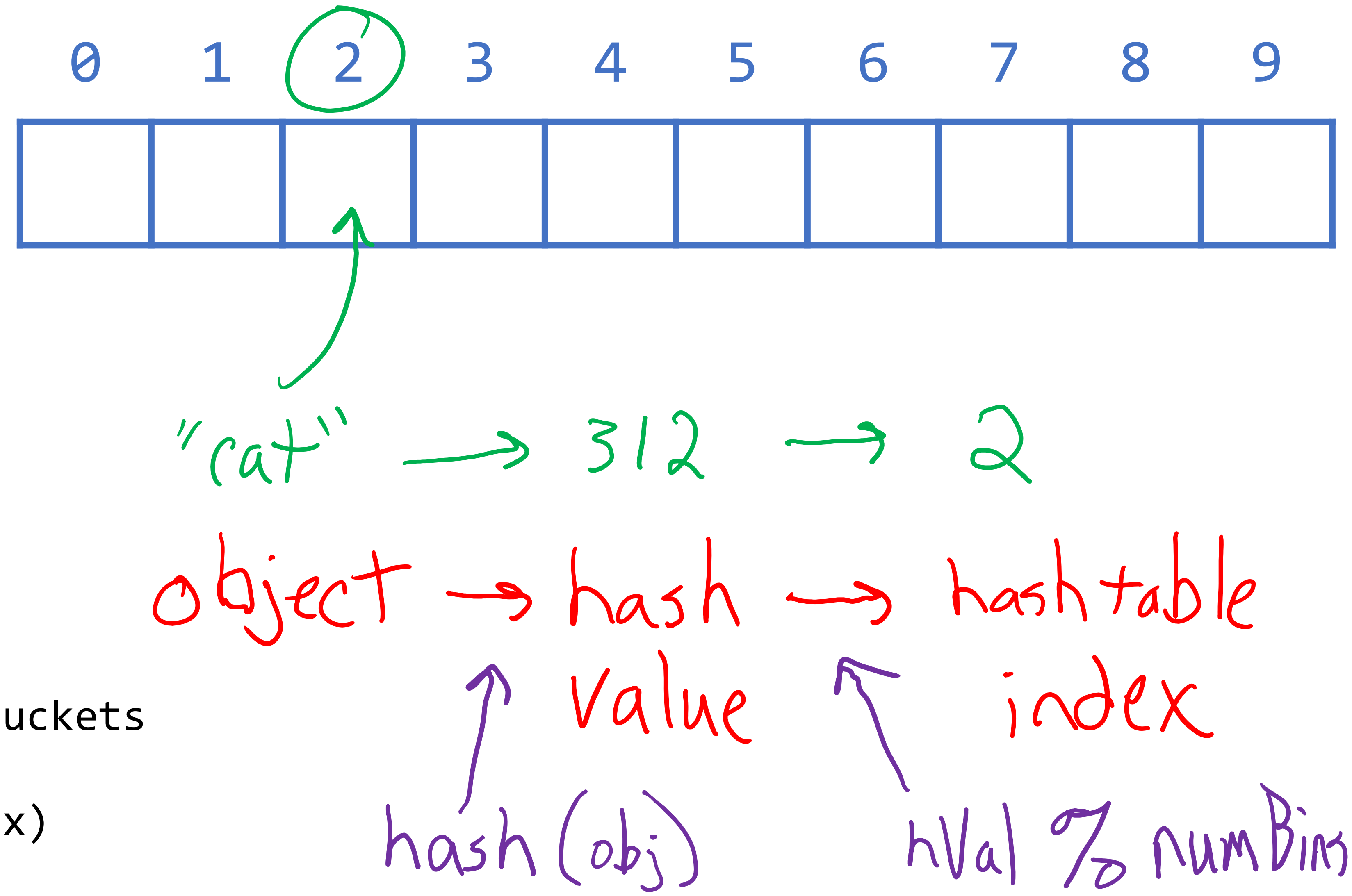
```
def myHash(s):
    val = 0
    for c in s:
        val += ord(c)
    return val

s = "cat"

numBuckets = 10
hVal = myHash(s)
bucketIndex = hVal % numBuckets

print(s, hVal, bucketIndex)
```

Hashtable (a list of length numBuckets)



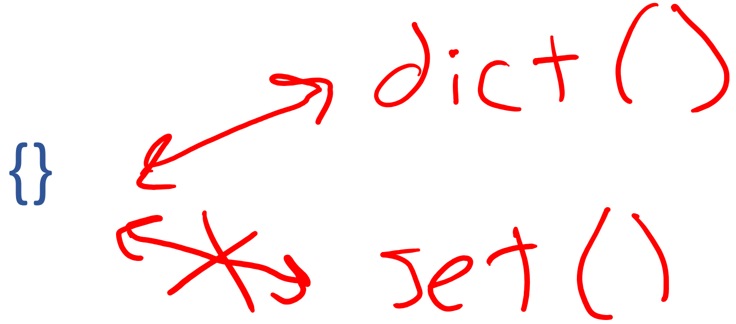
Dictionaries

Dictionaries

Map keys to values

vocab.get('ddd', 0)

Keys are stored like sets



{}

dict()

set()

{'a': 7}

{'a'}

if 'ddd' in vocab

safe

Key Error val = vocab['ddd']

Sets and Dictionaries Example

Vocab

Efficiency

Counting operations

Worksheet

Counting operations

Worksheet

$$\sum_{i=1}^{N-1} i$$

$$1 + 2 + \ldots + (N-1)$$

Counting operations

N is the size of the input data

- e.g. the length of an input list

The function f(N) is a measurement or count of resources used based on N

- Often based on computation time needed, but can be related to other resources like space (memory) needed
- Measured in number of operations rather than time
    - Lots of reasons, e.g. easier to compare algorithms despite changes in computer speed
- Small details either ignored or estimated (because of big-O)

Big O

$$\log_{17} N = \frac{\log_2 N}{\log_2 17}$$

Describes asymptotic behavior of a function

Informally (for 15112):

- Ignore all lower-order terms and constants →

f(N) is in O(?)

A few examples:

- $\cancel{1}N^2 \cancel{- \cancel{2}N} \cancel{+ 25}$ is in $O(N^2)$
- $\cancel{30000}N^2 \cancel{+ \cancel{2}N} \cancel{- 25}$ is in $O(N^2)$
- $\cancel{0.000001}N^2 \cancel{+ \cancel{123456}N}$ is in $O(N^2)$
- $\cancel{10}\boxed{N \log_{17} N} + \cancel{23}\boxed{N} - \boxed{17}$ is in $O(N \log N)$

Common Function Families

Constant:	$O(1)$
Logarithmic:	$O(\log N)$
Linear:	$O(N)$
Loglinear:	$O(N \log N)$
Quadratic:	$O(N^2)$
Exponential:	$O(2^N)$

Previous Poll 2

Which of these needs to visit all N elements in the list data, assuming N = len(data)?

Select ALL that apply.

```
A. for x in data:                    O(N)
       print(x)
B. for i in range(len(data)):        O(N)
       print(x)
C. if x in data:                     O(N)
       print(“Found it”)
D. x = data[i]                       O(1)
E. x = max(data)
F. None of the above
```

Previous Poll 3

Which of these needs to visit all N elements in the set data, assuming N = len(data)?

Select ALL that apply.

A. for x in data: O(N)
    print(x)

B. ~~for i in range(len(data)):~~ N/A
    x = data[i]
    ~~print(x)~~

C. if x in data: O(1)
    print(“Found it”)

D. ~~x = data[i]~~

E. x = max(data) O(N)

F. None of the above

Poll 4

Which of these is O(N) for the dictionary `d`, assuming N = len(d)?
Note: all of these are either O(1) or O(N)

Select ALL that apply.

A. `for key in d:` O(N)
      `print(d[key])`

B. `for i in range(len(d)):` O(N)
      `print(d[i])`

C. `if key in d:` O(1)
      `print(“Found it”)`

D. `x = d[key]` O(1)

E. `d[key] = x` O(1)

F. `None of the above`

Efficiency of Search and Sort

Example Lame Wordle

Poll 5

I’m thinking of a number between 1 and 64. After each guess, I’ll tell you if you’re correct or if my number is higher or lower.

$100 if you win. $0 if you lose.

How many guesses do you want to buy, $1 each?

A: 6

B: 7

C: 32

D: 64

Guess a Number: Binary Search

I’m thinking of a number between 1 and N. After each guess, I’ll tell you if you’re correct or if my number is higher or lower.

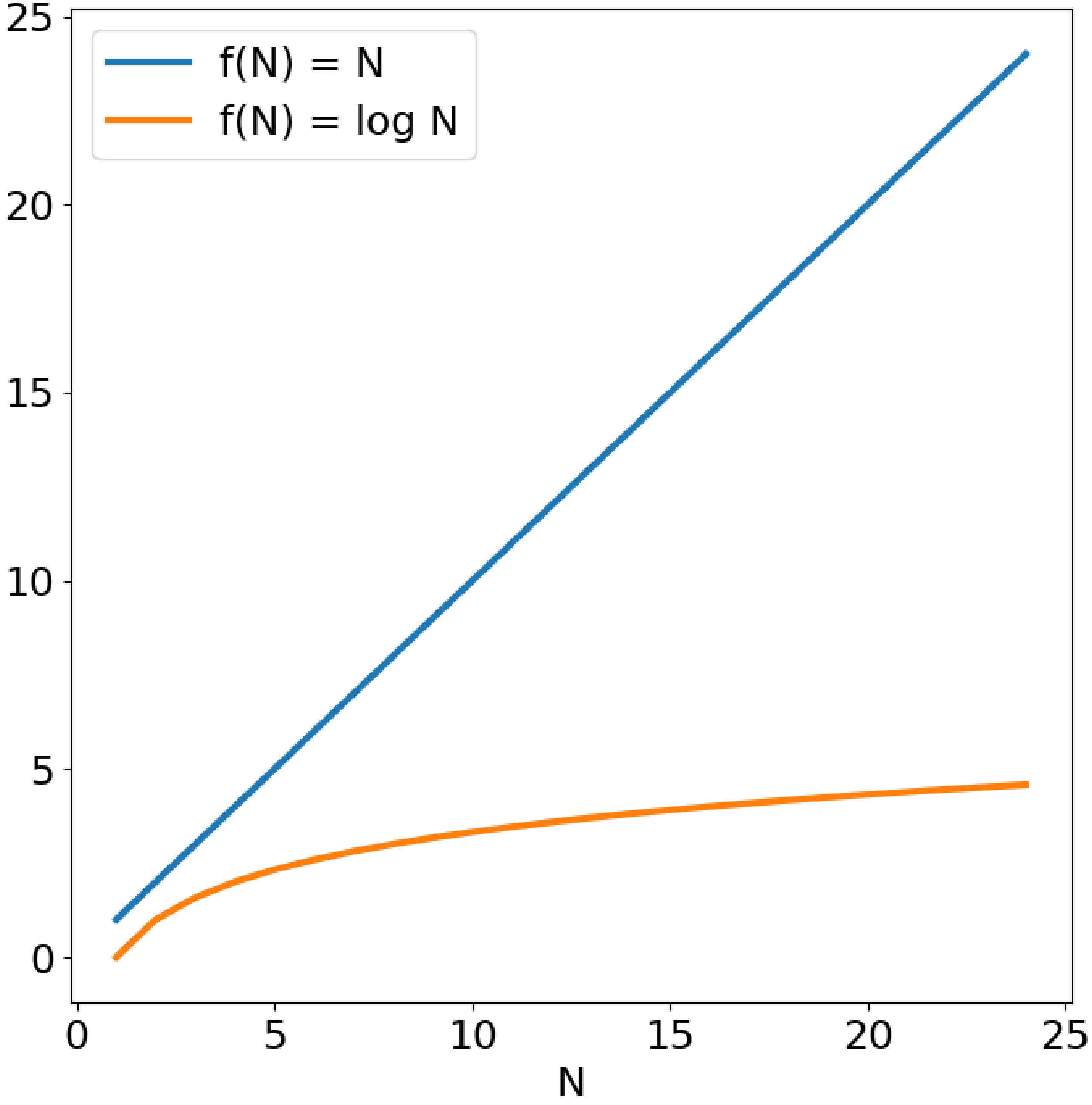
What is the maximum number of guesses you’ll need to play this game?

N	10	100	1,000	10,000	100,000	1,000,000	10,000,000
$\log_2 N$	3.3	6.6	10.0	13.3	16.6	19.9	23.3
$\lfloor \log_2 N \rfloor + 1$	4.0	7.0	11.0	14.0	17.0	20.0	24.0

Linear vs Binary Search

Linear search: $O(N)$

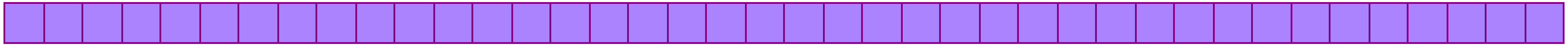
Binary search: $O(\log N)$



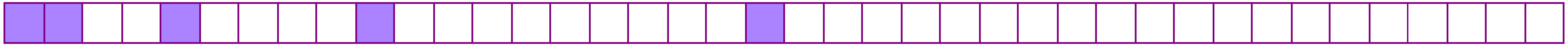
Linear vs Binary Search

Linear search: $O(N)$

N = 40



Binary search: $O(\log N)$



Common Function Families

Constant: $O(1)$

Logarithmic: $O(\log N)$

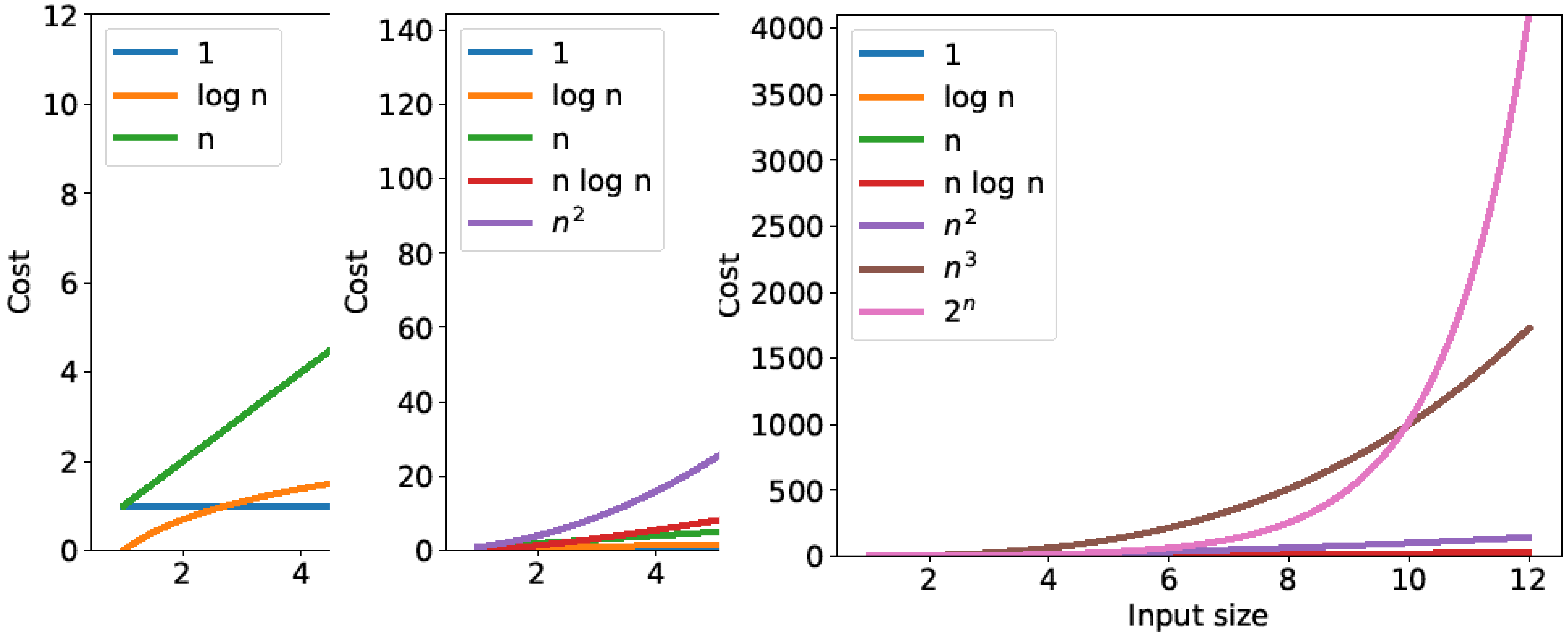
Linear: $O(N)$

Loglinear: $O(N \log N)$

Quadratic: $O(N^2)$

Exponential: $O(2^N)$

Complexity Classes



Efficiency of Sorting Algorithms

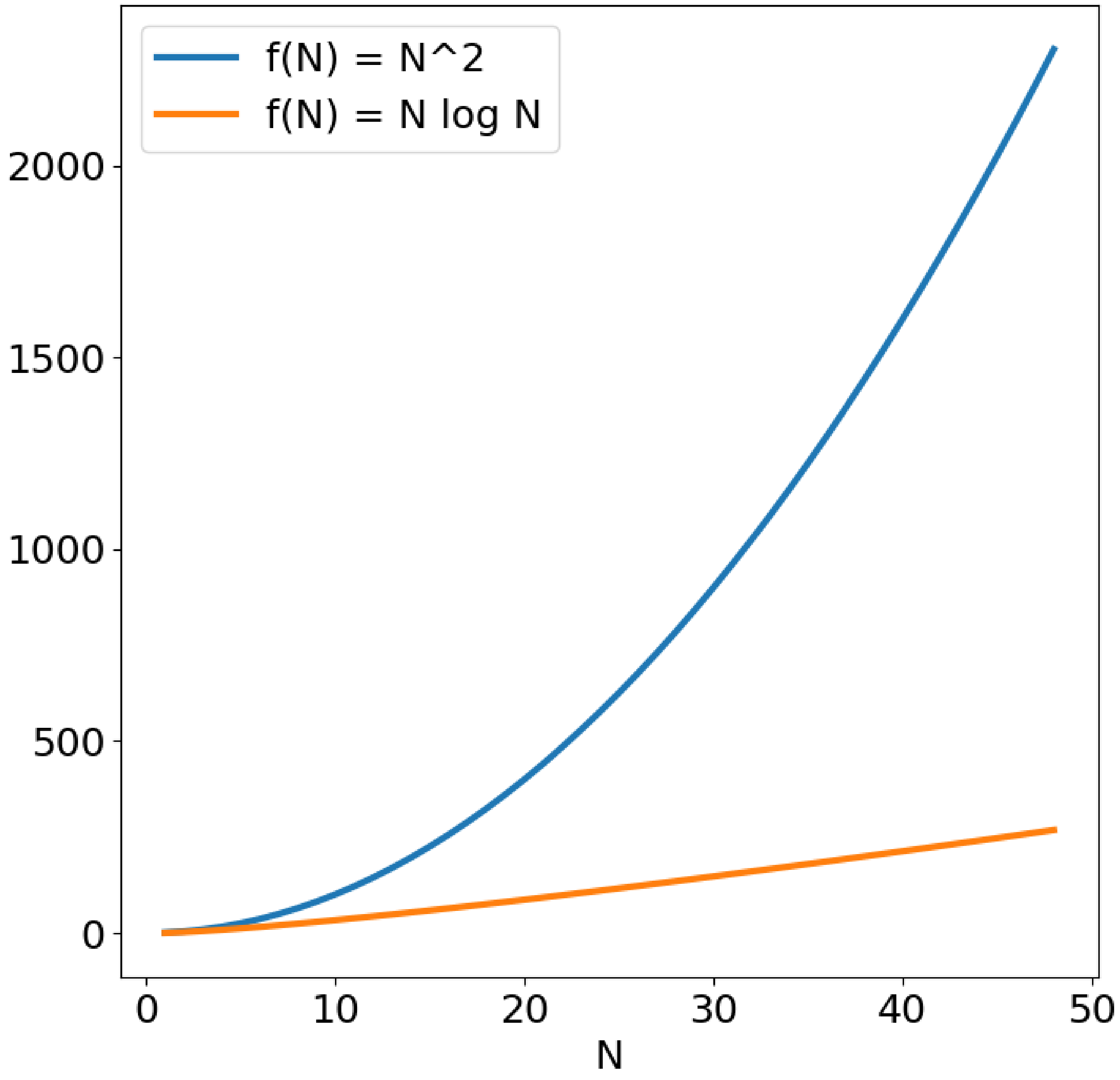
Sorting algorithms

xSortLab

http://math.hws.edu/eck/js/sorting/xSortLab.html

Selection sort: $O(N^2)$

Merge sort: $O(N \log N)$



Sorting algorithms

Selection sort: $O(N^2)$

Loop

- Find max in unsorted region
- Swap max with value at the end of the unsorted region
- Shrink unsorted region by 1

Merge sort: $O(N \log N)$

Sorting algorithms

Merge sort: $O(N \log N)$

Merge concept:

Assume you had two piles that were already independently sorted.

Could you shuffle them together into one sorted pile in O(N)?