From Bubble Sort to Quick Sort

Hui Chen
Department of Engineering & Computer Science
Virginia State University
February 10, 2017

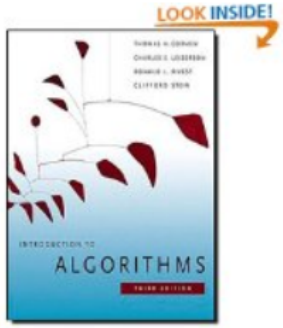
Acknowledgement

- This presentation is inspired by the following resources
  - The CS Unplugged project
    - http://csunplugged.org/
  - The OpenDSA project
    - https://opendsa-server.cs.vt.edu/
  - The Runestone Interactive project
    - Problem Solving with Algorithms and Data Structures using Python
    - http://interactivepython.org/runestone/static/pythonds/index.html

Where do we sort?

- Where are we ordering a list of objects?

‹ Return to product information | Have one to sell? | Every purchase on Amazon.com is protected by an A-to-z guarantee. | Feedback on this



Introduction to Algorithms, 3rd Edition (MIT Press) (Hardcover)

by Thomas H. Cormen, et al.

432 customer reviews | Share

Access codes and supplements are not guaranteed with used items.

Refine by Clear all

Shipping
- [ ] Prime
- [ ] Free shipping

Condition
- [x] New
- [ ] Rental
- [ ] Used
  - [ ] Like New
  - [ ] Very Good
  - [ ] Good
  - [ ] Acceptable

Price + Shipping	Condition (Learn more)	Delivery
$49.94 + $3.99 shipping + $0.00 estimated tax	**New**	• **Arrives between** March 3-23. • Ships from India. Learn more abou import fees and international shipp time. • Shipping rates and return policy.
$49.95 + $3.99 shipping + $0.00 estimated tax	**New** Delivery within 4 to 7 business days, great customer service. do ... » Read more	• **Arrives between** March 3-23. • Shipping rates and return policy.
$78.00 + $3.99 shipping + $0.00 estimated tax	**New**	• **Arrives between** February 10-28. • Ships from MN, United States. • Shipping rates and return policy.
$78.56	**New**	• **Arrives between** Feb. 13 - Mar. 1

Google

sorting

All Images Videos Books News More Settings Tools

About 117,000,000 results (0.66 seconds)

sort

/sôrt/

verb
gerund or present participle: **sorting**

1. arrange systematically in groups; separate according to type, class, etc.
   "she **sorted out** the clothes, some to be kept, some to be thrown away"
   synonyms: classify, class, categorize, catalog, grade, group; More
2. resolve (a problem or difficulty).
   "the teacher helps the children to **sort out** their problems"
   synonyms: resolve, settle, solve, fix, work out, straighten out, deal with, put right, set right, rectify, iron out; More

Translations, word origin, and more definitions

Feedback

Sorting algorithm - Wikipedia

https://en.wikipedia.org/wiki/**Sorting**_algorithm

A **sorting** algorithm is an algorithm that puts elements of a list in a certain order. The most-used orders are numerical order and lexicographical order.

Quicksort · Merge sort · Bubble sort · Selection sort

Watch a video

- About 1'30"
- An important figure was asked a question relevant to today's discussion. Let's see what he had to say.
- The video is at,
  - https://youtu.be/k4RRi_ntQc8

Google

sorting

All Images Videos Books News More Settings Tools

About 117,000,000 results (0.66 seconds)

sort

/sôrt/

verb

gerund or present participle: **sorting**

1. arrange systematically in groups; separate according to type, class, etc.
   "she **sorted out** the clothes, some to be kept, some to be thrown away"
   synonyms: classify, class, categorize, catalog, grade, group; More
2. resolve (a problem or difficulty).
   "the teacher helps the children to **sort out** their problems"
   synonyms: resolve, settle, solve, fix, work out, straighten out, deal with, put right, set right, rectify, iron out; More

Translations, word origin, and more definitions

Feedback

Sorting algorithm - Wikipedia

https://en.wikipedia.org/wiki/**Sorting**_algorithm

A **sorting** algorithm is an algorithm that puts elements of a list in a certain order. The most-used orders are numerical order and lexicographical order.

Quicksort · Merge sort · Bubble sort · Selection sort

Objective

- To be able
  - to explain and implement bubble sort and quick sort
  - to learn to evaluate sorting algorithms
    - to explain time complexity of bubble sort and quick sort, and
  - to gain insight of the divide-and-conquer strategy

How do we sort?

- Let's try to understand what bubble sort is.

Sorting objects by weight

- How do we sort a few objects by weight given a balance scale?

How many steps do we take?

- How many comparisons do we take?
- How many exchanges do we take?
- What if we have more objects to sort?

Why do we call it bubble sort?

- Let's watch a demo

Can we sort faster?

- Can we sort using less steps?
  - less comparisons
  - less exchanges
- Consider a divide-and-conquer strategy.
  - Choose an object, and use it to divide the objects into two halves

Sorting objects more quickly

- Let's apply the divide-and-conquer approach using a balance scale

How many steps do we take?

- Bubble sort and quick sort
  - How many comparisons do we take?
  - How many exchanges do we take?
- What if we have more objects to sort?

Sorting: a summary

- Data structure
  - Lists
- Basic operations
  - Comparison and exchange
- Two algorithms
  - Bubble sort
  - Quick sort

Bubble Sort

```
def bubble_sort(objects):
    for num_elems in range(len(objects)-1, 0, -1):
        # bubble the greatest element up
        for i in range(num_elems):
            if objects[i] > objects[i+1]:
                objects[i], objects[i+1] = objects[i+1], objects[i]
```

Quick Sort

```
def quick_sort(objects):
    pivot = objects[0]          # pick a pivot
    # divide the objects into two halves, lighter & heavier
    lighter = [elem for elem in data_list[1:] if elem <= pivot]
    heavier = [elem for elem in data_list[1:] if elem > pivot]
    # apply the same strategy to the two halves
    lighter = quick_sort(lighter)
    heavier = quick_sort(heavier)
    return lighter + [pivot] + heavier
```

Do some experiments

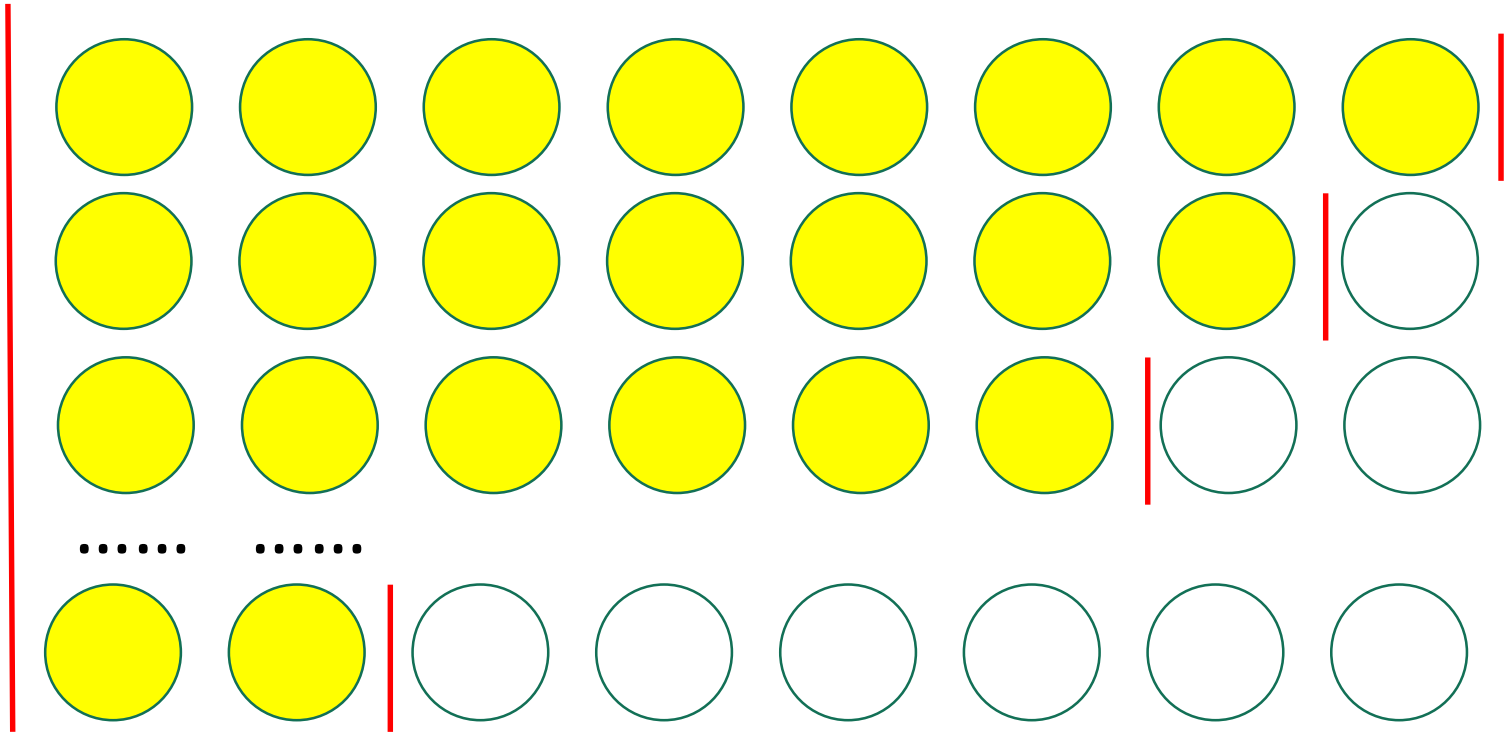
- Implement the algorithms in Python
- Measure running time used to sort lists of different sizes
- Observe the relationship between the running time and the size of the lists
- Code
  - https://goo.gl/QA3pNO (require only browser)
  - Github repository: https://goo.gl/phTjrS

Graph of what function is linear in log-log scale?

- How about $f(n) = n^2$?
- $\log f(n) = \log n^2 = 2 \log n$
- Let $y = \log f(n)$, $x = \log n$, we get
- $Y = 2x$

Analysis of Bubble sort

- The number of comparison
  - (n-1) + (n-2) + (n-3) + …+ 1 = n(n-1)/2

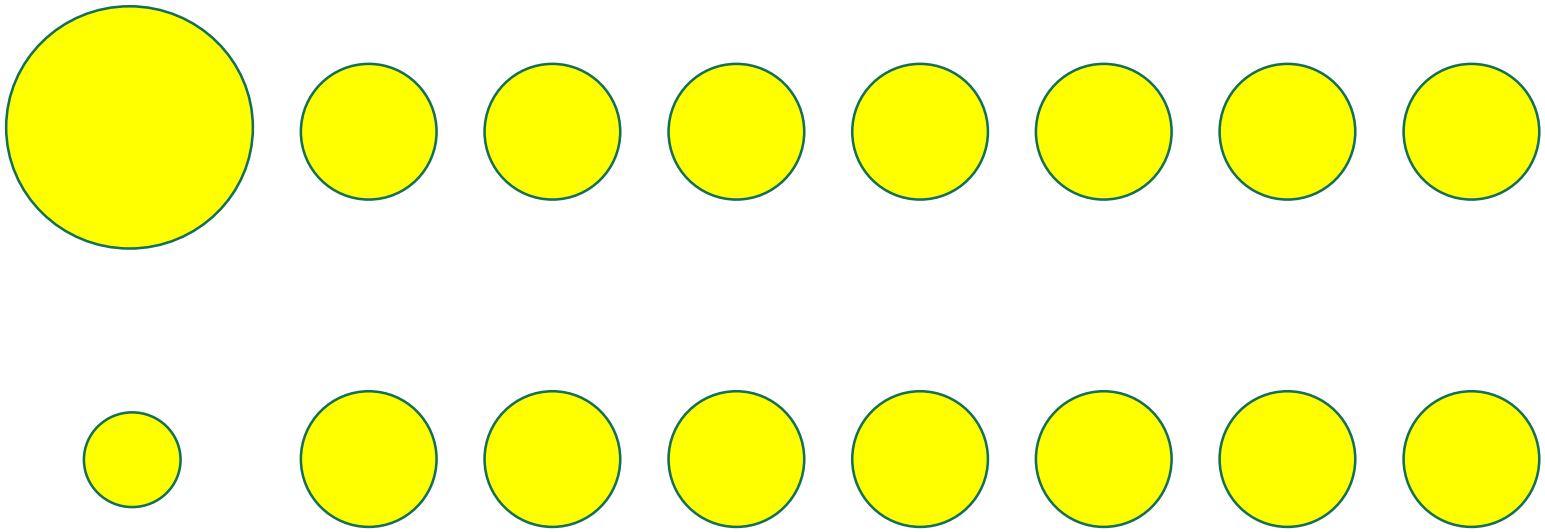


Analysis of Bubble sort

- How about the number of exchanges?

Analysis of Bubble sort

- How about the number of exchanges?
- It depends! See the following two cases.



How about analysis of Quick sort?

Discussion on running time

- Does the bubble sort always take the same amount of running time given the list of the same size?
- Does the quick sort always take the same amount of running time given the list of the same time?

Discussion on running time

- Does a sorting algorithm always take the same amount of running time given the list of the same size?
- Consider 3 cases
  - Ordered randomly
  - Already sorted in descending order
  - Already sorted in ascending order

Running time

- What is the relationship between running time and the size of list?
  - Can you predict the running time given the size of a list?
  - Orders of growth
  - Big-O, Big-Θ notations

Discussion on choice of pivot

- For quick sort, does the pivot have to be the first element?

Discussion on divide-and-conquer

- For sorting, is there any other divide-and-conquer approach you can think of?

Sorting algorithm evaluation

- How do we evaluate sorting algorithms?
  - Running time
  - Memory requirement
  - Stability
- How do we choose a sorting algorithm?

Exercise

- Measure running time for both Bubble sort and Quick sort
- Consider 3 cases
  - Ordered randomly
  - Already sorted in descending order
    - my_list = list(range(1000, 0, -1))
  - Already sorted in ascending order
    - my_list = list(range(0, 1000, 1))

Thank you!

- Discussed Bubble sort & Quick sort
  - They are comparison-exchange based sorting algorithms
  - They have significantly different time complexity
  - You shall see more applications of the divide-and-conquer strategy
- Some exercises for you!
- Do you have any questions?

huichen@ieee.org