We'll be starting shortly!

To help us run the workshop smoothly, please kindly:

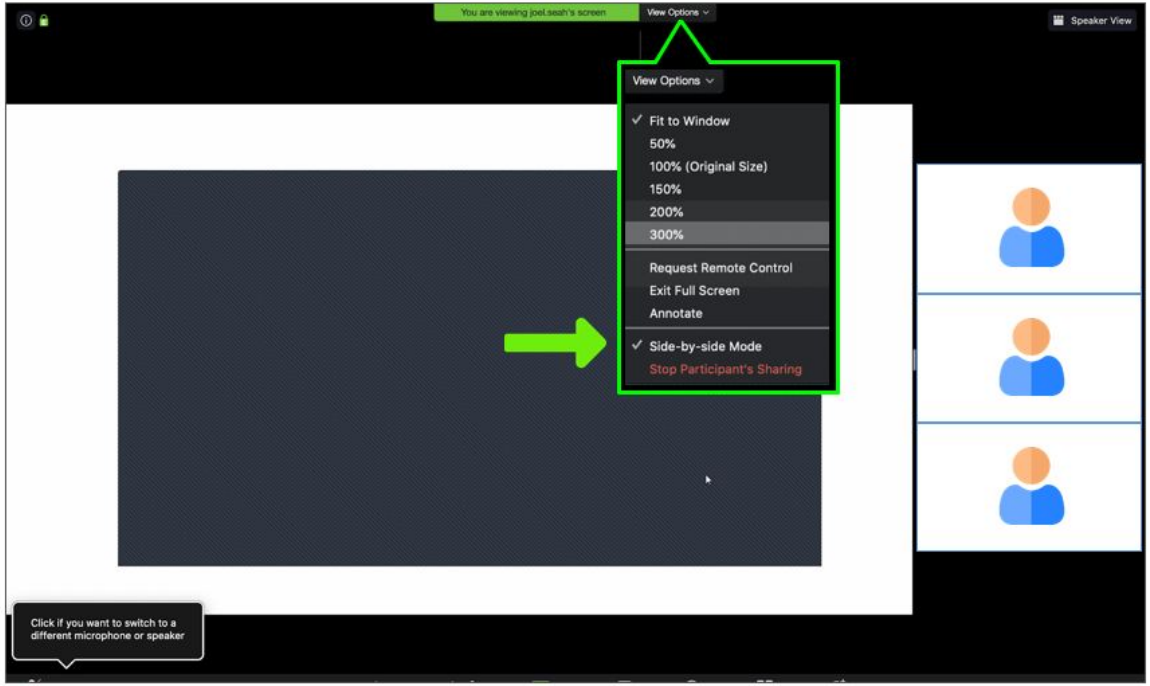
- Submit all questions using the Q&A function
- If you have an urgent request, please use the "Raise Hand" function

Thank you!

Vicohub

Shopee CODE LEAGUE 2020

Using Zoom: People & Slides



Side-By-Side Mode

- When sharing screen (slide share)
- With small thumbnails of people on the sidebar

STEPS:

1. View Options
2. Side-By-Side Mode

Sorting Algorithms

Dr. Quang Hieu VU

Vicohub

Introduction

- Sorting refers to arranging a set of data into some logical order
  - Given a sequence of items, each associated with a given key value. The problem is to rearrange the items so that they are in an increasing (or decreasing) order by key

- Sorting is among the most basic problems in algorithm design
- It's also an important factor that contributes to efficient search

Vicohub

Shopee CODE LEAGUE 2020

Introduction

- Sorting can be divided into
  - Internal sorting: if all the data that is to be sorted can be adjusted at a time in main memory
  - External sorting: when the data to be sorted cannot be accommodated in the memory at the same time and some has to be kept in auxiliary memory

- In reality, we can combine both internal sorting and external sorting to leverage advantages of both approaches

Vicohub

Shopee CODE LEAGUE 2020

Bubble sort

- Sometimes referred to as sinking sort
  - This algorithm repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order. The process stops when the list is in order.
- The algorithm, which is a comparison sort, is named for the way smaller or larger elements "bubble" to the top of the list.
- Although the algorithm is simple, it is too slow and impractical for most problems.
  - Bubble sort can only be practical if the input is in mostly sorted order with a few out-of-order elements.

Vicohub

Shopee CODE LEAGUE 2020

Bubble sort example

An example to sort [5, 1, 4, 2, 8]

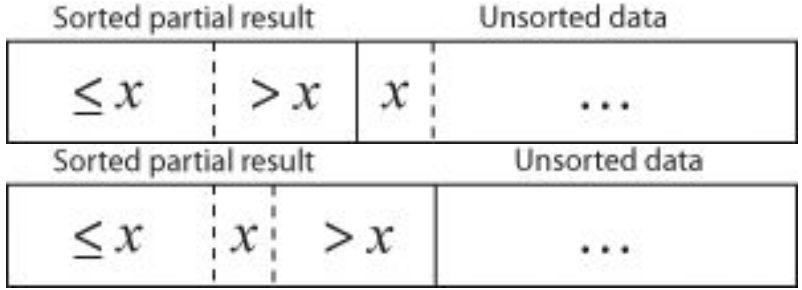
- [**5, 1**, 4, 2, 8] -> [**1, 5**, 4, 2, 8]
- [1, **5, 4**, 2, 8] -> [1, **4, 5**, 2, 8]
- [1, 4, **5, 2**, 8] -> [1, 4, **2, 5**, 8]
- [1, **4, 2**, 5, 8] -> [1, **2, 4**, 5, 8]
- The list is now in order and the process stops

Vicohub

Shopee CODE LEAGUE 2020

Insertion sort

- Sorting is typically done in-place, by iterating up the array, growing the sorted list behind it.
  - At each array-position, it checks the value there against the largest value in the sorted list (from the last position of the already sorted list). If larger, it leaves the element in place and moves to the next. If smaller, it finds the correct position within the sorted list, shifts all the larger values up to make a space, and inserts into that correct position.



Vicohub

Shopee CODE LEAGUE 2020

Insertion sort example

An example to sort [3, 7, 4, 9, 5, 2, 6, 1]

- **3***, 7, 4, 9, 5, 2, 6, 1
- 3, **7***, 4, 9, 5, 2, 6, 1
- 3, **4***, 7, 9, 5, 2, 6, 1
- 3, 4, 7, **9***, 5, 2, 6, 1
- 3, 4, **5***, 7, 9, 2, 6, 1
- **2***, 3, 4, 5, 7, 9, 6, 1
- 2, 3, 4, 5, **6***, 7, 9, 1
- **1***, 2, 3, 4, 5, 6, 7, 9

Vicohub

Shopee CODE LEAGUE 2020

Selection sort

- The algorithm divides the input list into two parts
  - The sublist of items already sorted, which is built up from left to right at the front (left) of the list
  - The sublist of items remaining to be sorted that occupy the rest of the list
- Initially, the sorted sublist is empty and the unsorted sublist is the entire input list. The algorithm proceeds by finding the smallest (or largest, depending on sorting order) element in the unsorted sublist, exchanging (swapping) it with the leftmost unsorted element (putting it in sorted order), and moving the sublist boundaries one element to the right

Vicohub

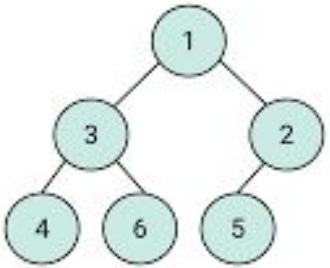
Shopee CODE LEAGUE 2020

Selection sort example

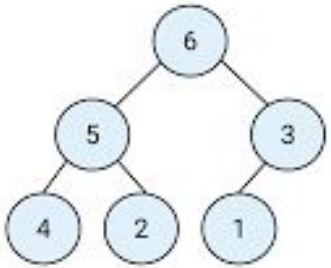
Sorted sublist	Unsorted sublist	Least element
[]	[11, 25, 12, 22, 64]	11
[11]	[25, 12, 22, 64]	12
[11, 12]	[25, 22, 64]	22
[11, 12, 22]	[25, 64]	25
[11, 12, 22, 25]	[64]	64
[11, 12, 22, 25, 64]	[]	

Heap sort

- The Heapsort algorithm leverages a max/min heap to sort decreasing/increasing
  - Max/min heap: parent nodes keep higher/smaller values compared to children nodes
  - Children nodes at the same level can keep arbitrary values



Min heap



Max Heap

Vicohub

Shopee CODE LEAGUE 2020

Heap sort

- The algorithm first build a max/min heap
- Then it repeatedly do the following steps until the range of considered values is one value in length
  - Swap the first value of the list with the last value
  - Decrease the range of values considered in the heap operation by one
  - Shifting the new first value into its position in the heap. This repeats until the range of considered values is one value in length.
- An example to sort [6, 5, 3, 1, 8, 7, 2, 4]

Vicohub

Shopee CODE LEAGUE 2020

Heap sort example

An example to sort [6, 5, 3, 1, 8, 7, 2, 4]

Vicohub

Shopee CODE LEAGUE 2020

Quick sort

- Quicksort is a divide and conquer algorithm
  - Also called partition-exchange sort
- Quicksort first divides a large array into two smaller sub-arrays: the low elements and the high elements.
  - Pick an element, called a pivot, from the array.
  - Partitioning: reorder the array so that all elements with values less than the pivot come before the pivot, while all elements with values greater than the pivot come after it (equal values can go either way). After this partitioning, the pivot is in its final position. This is called the partition operation.
- Recursively apply the above steps to the sub-arrays

Vicohub

Shopee CODE LEAGUE 2020

Quick sort example

An example to sort [3, 7, 8, 5, 2, 1, 9, 5, 4], use the last one as pivot

- Select 4 as pivot: [3, 1, 2], **4**, [5, 8, 9, 5, 7]
- Select 2 as pivot: [1], **2**, [3], **4**, [5, 8, 9, 5, 7]
- Select 7 as pivot: [1], **2**, [3], **4**, [5, 5], **7**, [8, 9]
- Select 5 as pivot: [1], **2**, [3], **4**, **5, 5**, **7**, **8**, [9]
- Select 8 as pivot: [1], **2**, [3], **4**, **5, 5**, **7**, [8], **9**

Vicohub

Shopee CODE LEAGUE 2020

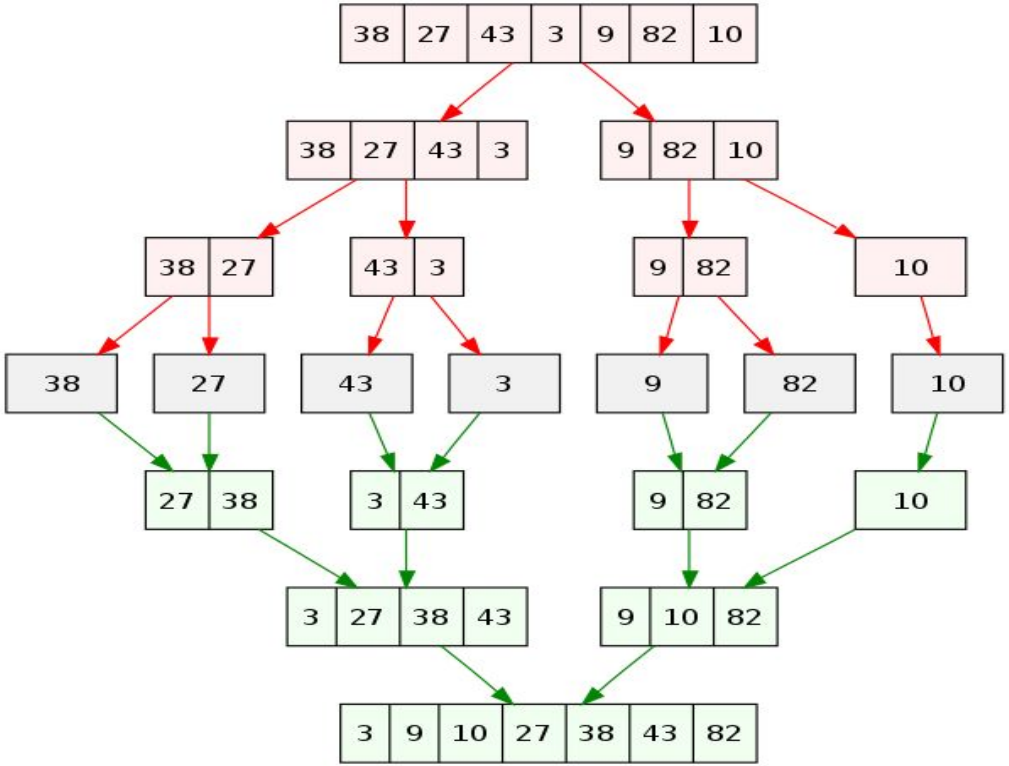
Merge sort

- Conceptually, a merge sort works as follows:
  - Divide the unsorted list into n sublists, each containing one element (a list of one element is considered sorted).
  - Repeatedly merge sublists to produce new sorted sublists until there is only one sublist remaining. This will be the sorted list.
- Merge sort naturally supports external sort
- Can also run in parallel

Vicohub

Shopee CODE LEAGUE 2020

Merge sort example



Vicohub

Shopee CODE LEAGUE 2020

Comparison of sort algorithms

Algorithm	Average	Best	Worst	Type
Bubble sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	Internal sort
Insertion sort	$O(n^2)$	$O(n)$	$O(n^2)$	Internal sort
Selection sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	Internal sort
Heap sort	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	Internal sort
Quick sort	$O(nlogn)$	$O(nlogn)$	$O(n^2)$	Internal sort
Merge sort	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	External sort

Your Feedback Matters!

https://techatshopee.formstack.com/forms/shopeecodeleague_workshopfeedbackform

Vicohub

Shopee CODE LEAGUE 2020