IT5003 Semester 2 2024/2025

Data Structures and Algorithms

Tutorial 02
Sorting, ADT, List
For Week 04 (Sat)/05 (Mon)

Document is last modified on: January 17, 2025

1 Introduction and Objective

Welcome back after the Chinese New Year 2025 break.

In this tutorial, we will discuss various Sorting algorithms and the concept of Abstract Data Type (ADT) that will appear several times throughout this course. We discuss List ADT today.

You are encouraged to review e-Lecture of `https://visualgo.net/en/array?slide=1` (this new visualization page is evolving quickly), `https://visualgo.net/en/sorting?slide=1` (IT5003 students can skip Radix Sort), and `https://visualgo.net/en/list?slide=1` for this tutorial.

2 Tutorial 02 Questions

Sorting

Q1). At `https://visualgo.net/en/array?slide=2-7`, Prof Halim mentions a few (not exhaustive) array (also known as list in Python) applications that become easier/more efficient if the underlying array is sorted. Now, we will quickly discuss application a few of these application 1-7 in algorithmic level only.

Sorting, Mini Experiment

Q2). Please use the 'Exploration Mode' of `https://visualgo.net/en/sorting` to complete the following table. You can use 'Create' menu to create input array of various types. You will need to fully understand the individual strengths and weaknesses of each sorting algorithms discussed in class in order to be able to complete this mini experiment properly.

For example, on random input, Optimized (Opt) Bubble Sort that stops as soon as there is no more swap inside the inner loop runs in $O(N^2)$ but if we are given an non-decreasing numbers as input, that Optimized Bubble Sort can terminate very fast, i.e., in $O(N)$.

Note that N-d and N-i means Non-decreasing (increasing or equal) and Non-increasing (decreasing or equal), respectively.

Note that Many Duplicates test cases include All Equal test cases.

Note also that the term 'Nearly sorted' can have multiple definitions and we will discuss this in class.

Input type → ↓ Algorithm	Random	Sorted		Nearly Sorted		Many Duplicates
		N-d	N-i	N-d	N-i	
(Opt) Bubble Sort	$O(N^2)$	$O(N)$				
(Min) Selection Sort					$O(N^2)$	
Insertion Sort		$O(N)$				
Merge Sort				$O(N \log N)$		
Quick Sort		$O(N^2)$				
(Rand) Quick Sort	$O(N \log N)$					$O(N \log N)$
Counting Sort					$O(N)$	

Finding k-th Smallest Element in an Unsorted Array (Selection Algorithm)

Q3). We will revisit the concept of QuickSort's partitioning algorithm (please review `https://visualgo.net/en/sorting?slide=12-2` to slide 12-7) and combine it with a binary search-like algorithm on an unsorted array to find the k-th smallest element in the array efficiently. In this tutorial, we will spend some time discussing the details. Your job before attending this tutorial is to read this algorithm from the Internet: `http://en.wikipedia.org/wiki/Quickselect`, or if you have Competitive Programming 4 (or earlier) book, read about it in Section 2.3.4 (the earlier part of Order Statistics Tree).

Abstract Data Type (ADT)

Q4). Please self-read List ADT and its common operations: `https://visualgo.net/en/list?slide=2-1`. Now compare it with the sample (fixed-size array) implementation discussed back in last Tutorial Tut01 and if we replace that fixed-size array with Java ArrayList implementation instead. What are the concepts of ADT that you have understood by now? We will discuss List ADT in more details in Lecture when we discuss 3 related ADTs: Stack, Queue, Deque.

Hands-on 2

TA will run the second half of this session with a few to do list:

- Do a sample speed run of VisuAlgo online quiz that are applicable so far, e.g., `https://visualgo.net/training?diff=Medium&n=5&tl=5&module=sorting`.

- Finally, live solve another chosen Kattis problem involving sorting.

Problem Set 2

We will end the tutorial with high level discussion of PS2 A+B (and optional C).