# Topic 1: Basic Knowledge

## Basic Concept

### Guidance

Data Structures and Algorithms Made Easy.pdf chapter 1

Data Structures and Algorithm Analysis.pdf chapter 2

### Practice

1. Why learn data structures and algorithms?
2. What are some common data structures and briefly describe them
3. What is the logical structure of data? Classify all data structures according to their logical structure
4. What is the physical structure of data? Classify all data structures according to their physical structure
5. What are the common operations of the program? and briefly describe them. eg.insert
6. Which are the basic characteristics of the algorithm?
7. What aspects can be used to judge the quality of an algorithm?
8. What is time/space complexit of an algorithm? Why need care more about the worst case or the best case?

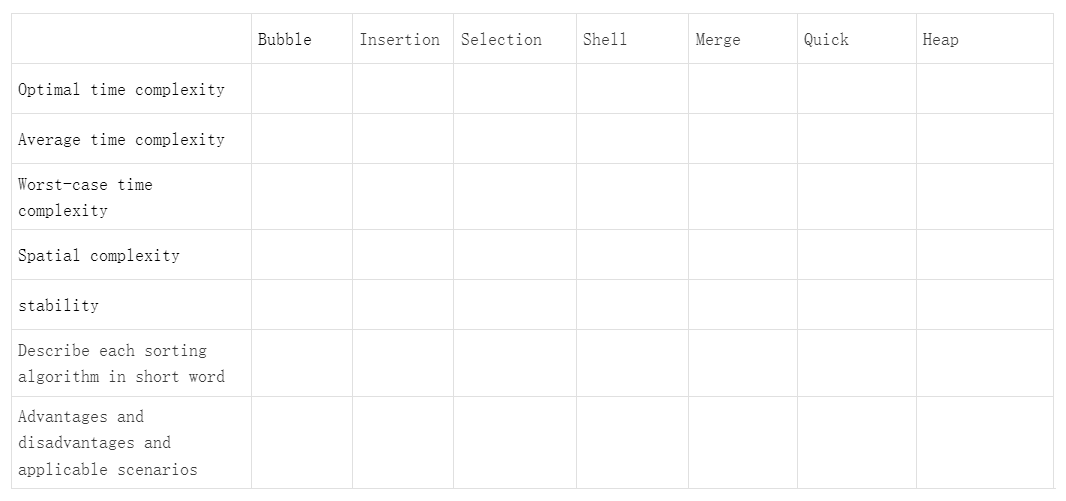
## Sorting

### Guidance

Data Structures and Algorithm Analysis.pdf->chapter 7

### Practice

1. After learning this topic, Summarize the different sorting algorithms



1. Data Structures and Algorithm Analysis.pdf 7.31
2. Data Structures and Algorithm Analysis.pdf 7.33
3. Takes the 3th largest element in a 10 numbers' array

There is a library function which name is qsort, Implement one of your own version named it my\_qsort and use it

## Stack & Queue

### Guidance

Data Structures and Algorithm Analysis.pdf 3.3, 3.4

### Practice

1. The sequence of stack out

There are 4 elements ABCD can be pushed into a stack. Which length is 4. Provide all possible sequences of popping elements from the stack.

For example, AB in, B out, C in, C out, D in, D out, A out. In this case, the sequence of popped elements is BCDA.

1. Implement Array-Based Stack and Queue

Stack functions：stack\_push stack\_pop stack\_size stack\_is\_empty stack\_is\_full stack\_top

Queues functions：que\_enqueue que\_dequeue que\_size que\_is\_empty que\_is\_full que\_front

Requirement: Queues can be recycled

1. Caculate the value of the expression

string：(4 - 2) \* 3 - 2 \* ((9 - 3) / 3)

# Topic 2: Linked List

## Type Of Linked List

### Guidance

Data Structures and Algorithms Made Easy.pdf chapter 3

* Singly linked lists.
* Doubly linked lists.
* Circular linked lists.
* Circular doubly linked lists.

### Practice

1. Draw the structure of these 4 types of linked list
2. Draw pictures(steps) to show that how to insert/delete a node to/from the top/middle/tail of these linked list
3. How to represent the state of the full and empty for these linked list

## Sorting

How to sort a singly/doubly linked list using different sorting algorithm without data swap?

If any sorting algorithm is not suitable for linked lists, give the explanation

## Stack & Queue

Implement Linked list based Stacks and Queues

Stack functions：stack\_push stack\_pop stack\_size stack\_is\_empty stack\_is\_full stack\_top

Queues functions：que\_enqueue que\_dequeue que\_size que\_is\_empty que\_is\_full que\_front

1. Singly Linked List
2. Doubly Linked List

## Open Source Library Use

### Guidance

Linux.Kernel.Development.3rd.Edition.pdf chapter 6 ->Linked List

### Practice

Use list\_\* function in file to write a program.

init, add, del, check, find, get last, foreach, reverse

# Topic 3: Searching

## Type Of Searching

### Guidance

Data Structures and Algorithms Made Easy.pdf chapter 11 13

* Compare and understand the differences between different search methods(best/worst/average time complexity)
* The difference between binary search and interpolation search

## BST

### Guidance

Data Structures and Algorithms Made Easy.pdf 6.11

* What scenarios need BST?
* How to construct BST, through Pre-order,In-order,Post-order and level-order?
* How to convert an ordinary binary tree into a binary search tree

## Hashing

### Guidance

Data Structures and Algorithms Made Easy.pdf 14

### Practice

1. What is hash table, hash function and collisions?
2. How many ways to handle collisions? What are their characteristics. Describe each collision handling in concise word
3. Implement the functions on 14.3 with Linear probing and separate chaining respectively

## uthash

### Guidance

code:<https://github.com/troydhanson/uthash>

doc:[https://troydhanson.github.io/uthash/userguide.html](https://troydhanson.github.io/uthash/userguide.html#_add_item)

### Practice

1. How does the hash function get the hash key?
2. Which method the project uses to handle collisions?
3. How is the new node added into the hash table?
4. What are knowledges that we have learned be used in the project?
5. How the HASH\_SORT implement the sorting?

## Hashing Used By Linux

### Guidance

The Linux TCP IP Stack - Networking for Embedded Systems.pdf chapter 10

How is hashing used for TCP and UDP in Linux kernel?