# Week1

# Week2

# Week3

# Week4

# Week5

# Week6

# Week7

# Week8

# Week9

## Pre-Sorting

### Searching

An array should be sorted first and then search can be performed if search activities are many. However, we should not sort and search if we only need to search a few times.

Complexity: O (NlogN) +O (N)/ O (logN) = O (NlogN) with sorting is NlogN and search is either linear or logN (binary search)

### Uniqueness Searching

### Activity Selection

### Computing the median

# Week10

## Greedy Approach

### Prim’s Algorithm

* Start with one randomly selected vertex and add it to a list
* Add a neighbor vertex that has minimum edge weight using min priority queue
* Check if there are smaller edge weight to a vertex in neighbor set of the current visiting vertex, and update it
* When all vertices added to tree, the algorithm is completed.

### Dijkstra’s Algorithm

* Initialize a set of all vertices (set D), and update their distance from the source to be infinity (unknown), with the source to be 0 so the algorithm knows where to start
* Initialize another set (set S) to store the current shortest path, initially should be empty
* Select a vertex that has the minimum shortest-path estimate that is not in S
* Add the vertex to S
* Update the distance estimation to the vertices that are not in S
* Repeat from step 3, until all vertices have been added to S

## Dynamic Programming

### Bottom-Up

### Top-Down