ClojureScript Algorithm Agenda List

# Maximum Of Three Numbers

**Problem:** Given three numbers x, y, and z, write a function to find the largest of the three.

# Swap Values

**Problem:** Given two numbers x and y, swap the values such that x holds the value originally in y, and y holds the value originally in x without using a temporary variable.

# Linear Search

**Problem Statement:** Given an array arr[] of n elements, write a function to search a given element x in arr[] with a O(n) time complexity.

**Solution Algorithm:** Start from the leftmost element of arr[] and one by one compare x with each element of arr[]. If x matches with an element, return the index. If x doesn’t match with any of elements, return -1.

# Binary Search

**Problem Statement:** Given a sorted array arr[] of n elements, write a function to search a given element x in arr[] with a O(log n) time complexity.

**Solution Algorithm:** Search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise narrow it to the upper half. Repeatedly check until the value is found or the interval is empty. Return element if found, -1 if not.

# Selection Sort

**Problem:** Write a function to sort a given an array arr[] of n elements.

**Solution Algorithm:** Find the minimum value in the list. Swap it with the value in the current position. Repeat this process for all the elements until the entire array is sorted.

# Bubble Sort

**Problem:** Write a function to sort a given an array arr[] of n elements.

**Solution Algorithm:** Repeatedly swap adjacent elements if they’re in wrong order.

# Insertion Sort

**Problem:** Write a function to sort a given an array arr[] of n elements.

**Solution Algorithm:** Compare each element with the elements to its left and find its right where it will be inserted. This forms a sorted portion on the left of the element, and an unsorted portion on its right.

# Quick Sort

**Problem:** Write a function to sort a given an array arr[] of n elements.

# Merge Sort

**Problem:** Write a function to sort a given an array arr[] of n elements.

**Solution Algorithm:** Divides input array in two halves, calls itself for the two halves and then merges the two sorted halves.

# pow(x,n)

**Problem:** Given two integers x and n, write a function to compute xn.