

CS 2202

Assignment in Algorithms and Complexities (Divide and Conquer)

Submitted by:

**Robert Michael Lim, BSCS – 3**

Submitted to:

**Ms. Christine F. Peña**

1. Divide and conquer

* There are many sorting algorithms out there. There are algorithms that solve it linearly, like bubble and selection sort. And, there is also divide and conquer algorithms. Divide and Conquer algorithms can be simply described by its name; to **divide** the problem into a number of subproblems and to **conquer** by solving the subproblems recursively. All in all, when it is combined, the number of sub problems can now be taken as one answer.

1. Mergesort and Quicksort

* Mergesort and Quicksort are the two common types of divide and conquer algorithms. On what I understood, Mergesort divides the whole array of numbers and solves them in an ascending order by sub-arrays, while Quicksort is used to partition the array into subarrays and conquers them by partitioning the subarrays to make it to an order.

1. An example of a divide and conquer algorithm

* A good example of a divide and conquer algorithm is the Binary Search. It uses recursive properties to use the concept of divide and conquer by comparing the middlemost/mean of the collection and sorting them.

Pseudocode:

BinarySearch(A[0..N-1], value, low, high) {

// invariants: value > A[i] for all i < low

value < A[i] for all i > high

if (high < low)

return not\_found // value would be inserted at index "low"

mid = (low + high) / 2

if (A[mid] > value)

return BinarySearch(A, value, low, mid-1)

else if (A[mid] < value)

return BinarySearch(A, value, mid+1, high)

else

return mid

}

Flowchart:

