	Manual de prácticas del Laboratorio de Estructuras de datos y algoritmos II	Código:	MADO-20
		Versión:	01
		Página	15/180
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Guía práctica de estudio 2

Estructura de datos y Algoritmos II

Algoritmos de Ordenamiento. Parte 2.

Objetivo: El estudiante conocerá e identificará la estructura de los algoritmos de ordenamiento *Quick Sort* y *Heap Sort*.

Actividades

- Implementar el algoritmo *Quick Sort* en algún lenguaje de programación para ordenar una secuencia de datos.
- Implementar el algoritmo *Heap Sort* en algún lenguaje de programación para ordenar una secuencia de datos.

Antecedentes

- Análisis previo de los algoritmos en clase teórica.
- Manejo de arreglos o listas, estructuras de control y funciones en Python 3.

Introducción

Quick Sort

Este algoritmo de ordenamiento al igual que *Merge Sort* sigue el paradigma divide y conquista por lo que en este documento se explican los tres procesos involucrados para ordenar una secuencia de números.

Para su descripción, la secuencia a ordenar está representada por un arreglo lineal o unidimensional. Los tres procesos son:

Divide: Se divide un arreglo *A* en 2 sub-arreglos utilizando un elemento pivote *x* de manera que de un lado queden todos los elementos menores o iguales a él y del otro los mayores. Figura 2.1.

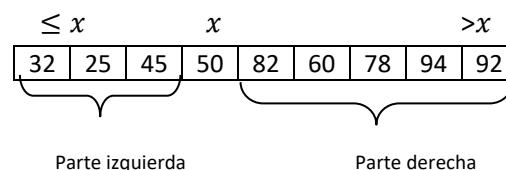



Figura 2.1

	Manual de prácticas del Laboratorio de Estructuras de datos y algoritmos II	Código:	MADO-20
		Versión:	01
		Página	14/180
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Guía Práctica de Estudio 2

Algoritmos de ordenamiento parte 2

Elaborado por:

M.I. Elba Karen Sáenz García

Revisión:

Ing. Laura Sandoval Montaña