**CS2001**

**Complexity**

**18/11/2020 190022143**

# Abstract

This report documents my work for the CS2001 W10 Practical, in which we explored how a search algorithm’s performance is affected by its pathological cases. The program was required to run a tested quicksort on pathological sorting cases and XXX.

# Introduction

**Quicksort**

The quick sort is a divide and conquer algorithm which picks a pivot element, partitions around that element, and recursively sorts the partitions. The pivot theoretically should be close to the median value, however, in an unsorted array the median value could be any one of them. For the purposes of this project, the final element is used as the pivot.

**Sortedness**

**https://link.springer.com/chapter/10.1007/BFb0038186**

There are a variety of ways in which one could define how sorted a list is. The general consensus is that Vladmir Estivill-Castro and Derick Wood give the best axiomatic definitions of measures of disorder.

Pathological cases

Algorithm testing

# Methodology

I decided to have different test class which capture the time it takes to run the tests. This was primarily for clean and modular code. It also allowed me to easily run the analysis by running the class or by using IntelliJ’s test runner. The result, which was stored in a CSV, was then put into a pandas data frame for analysis.

Long data types used as these are returned by getting the difference between start and end times.

# Results

## Evaluation and Conclusion