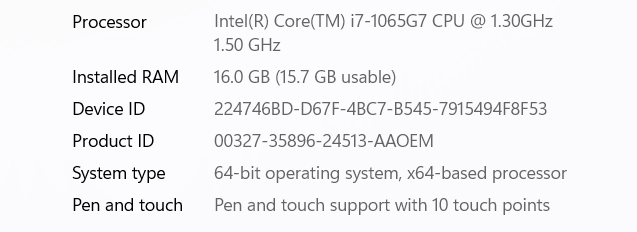
REPORT: SORTING

# Information

|  |  |
| --- | --- |
| Full Name | Student ID |
| Trần Nguyễn Minh Hoàng | 22127131 |
| Đỗ Đình Hải | 22127095 |
| Phạm Thành Đạt | 22127064 |
| Đỗ Phan Tuấn Đạt | 22127057 |

# Introduction

## Specifications of device used for sorting



# Algorithm presentation

## Selection sort

## Insertion sort

## Bubble sort

## Shaker sort

## Shell sort

## Heap sort

## Merge sort

## Quick sort

### Idea

Divide the given array into two sections using a partitioning element called a pivot. The division performed is such that all elements to the left side of the pivot are smaller than the pivot, and all elements to the right side of the pivot are greater than the pivot. Pivot element can be first element, last element, middle element, or even a random element in the array.

### Step-by-step description

Below is the algorithm for quick sort using the last element of the array as the pivot:

Step 1 − Choose the highest-index element as Pivot.

Step 2 − Take another variable to indicate Low (lowest index - 1).

Step 5 – Starting from index i = Low + 1, while i < index of Pivot, compare i-th element with Pivot

Step 5.1 – If value of i-th element is smaller than Pivot, increase Low by 1 and swap i-th element with Low-th element.

Step 5.2 – Increase I by 1 and return to step 5.1.

Step 6 – Swap (i+1)-th element with Pivot. The array is now divided into two sub-arrays. The left sub-array consists of all elements smaller than Pivot, and the right sub-array consists of all elements greater (or equal, if any) elements than Pivot.

Step 7 – Recursively perform the algorithm, starting from Step 1, for each sub-array.

When the process completely terminates, the resulting array will be a sorted array.

### Complexity evaluation

* Time complexity:
  + Best case: O(n\*logn)
  + Average case: O(n\*logn)
  + Worst case: O(n^2)
* Space complexity: O(1)

### Variants/Improvements

Quick sort can be sped up significantly by choosing the median of several elements as the Pivot.

## Counting sort

## Radix sort

## Flash sort

# Referenced resources

* <https://www.ques10.com/p/65800/write-quicksort-algorithm-using-last-element-as-pi/>
* <https://www.geeksforgeeks.org/quick-sort/>
* [Analysis of quicksort (article) | Quick sort | Khan Academy](https://www.khanacademy.org/computing/computer-science/algorithms/quick-sort/a/analysis-of-quicksort)