

# CPSC 2150 – Algorithms and Data Structures II

## Lab6: Binary Tree - Heapsort

Total - 30 Marks

---

### Learning Outcomes

- Design and develop an appropriate binary tree
- Implement heap structure using an array
- Design and implement the Heapsort
- Analyzing the heapsort
- Program with C++

### Resources

- Chapter 8 of the text book

### Description

This is a practice on implementing the heap structure and heapsort algorithm followed by analyzing it in terms of space and time complexity. For sake of simplicity consider an array of  $n$  integer elements, `data`, to be sorted in ascending order.

Implement the `heapSort` using a max-heap data structure.

- [20 marks]** Write a function named `heapSort()` that given a list of integers, sorts them in ascending order using the `heapSort` algorithm (**`heapSort.cpp`**).
- [5 marks]** Calculate the time complexity of `heapSort()` function (**`answers.pdf`**).
- [5 marks]** Calculate the space complexity of `heapSort()` function (**`answers.pdf`**).
- [5 bonus marks]** Do you think using a min-heap rather than a max-heap in your `heapSort()` where data is still sorted in ascending order would change the efficiency of the algorithm (**`answers.pdf`**).

### Submit to D2L

Make a **zip file** named **`StudentNumber-lab6.zip`** including all related files by the end of the lab time. For example if your student number is 10023449, the submitted file must be named as **`10023449-lab6.zip`**.