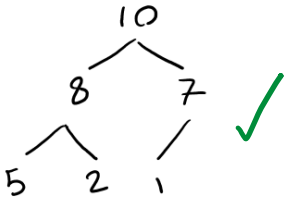
Exam 2:

Due Date: **Sunday, November 15, 2020, at 11:59pm.**

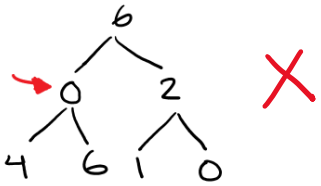
This exam contains two problems asking multiple questions. Please answer each question in detail with clear explanation. :)

Problem 1. A random complete binary tree in an array format is given to you. Develop an algorithm to verify whether the array is a max-heap or not.

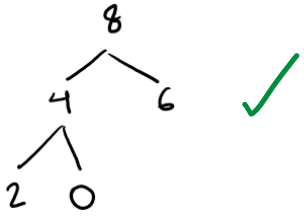
Example 1: *Input*: a = [10, 8, 7, 5, 2, 1] ➔ *Output*: "This is a max-heap."



Example 2: *Input*: a = [6, 0, 2, 4, 6, 1, 0] ➔ *Output*: "This is NOT a max-heap."



Example 2: *Input*: a = [8, 4, 6, 2, 0], ➔ *Output*: "This is a max-heap."



A. How would you decide if the array is a max-heap? (**Note**: If you have multiple answers in mind, break them apart and explain each one separately.) **Explain each solution/algorithm in detail**.

B. Write the pseudocode for the best algorithm you came up with.

C. Implement your answer using any programming language you want to.

D. What is the time complexity of your answer? **Explain in detail and show all the work**. (**Note**: If possible, break your code/pseudocode to different parts, calculate the runtime for each step and then try to calculate the total running time based on that.)

Problem 2. A *sorted* array of size *n* and a random array of size *k* are given to you (***k<n***). Develop an algorithm to decide whether the smaller random array is a subset of the larger sorted array. (**Note**: There are **no** repetitions in any of the arrays.)

Example 1: *Input*: a = [2, 3, 5, 7, 8, 10, 15], b = [2, 10 , 5] ➔ *Output*: "b is a subset of a."

Example 2: *Input*: a = [0, 2, 4, 8, 9, 12, 13, 15, 24], b = [6, 0 , 4, 19, 35] ➔ *Output*: "b is **not** a subset of a."

A. How would you decide if *b* is a subset of *a*? (**Note**: If you have multiple answers in mind, break them apart and explain each one separately.) **Explain each solution/algorithm in detail**.

B. Write the pseudocode for the best algorithm you came up with.

C. Implement your answer using any programming language you want to.

D. What is the time complexity of your answer? **Explain in detail and show all the work**. (**Note**: If possible, break your code/pseudocode to different parts, calculate the runtime for each step and then try to calculate the total running time based on that.)