BST and Graph

(For task 01 & 02, you may want to create a binary search tree **(BST)** manually.

Task 03 is a **hand-written in-class** submission task)

1. Writea method that **RECURSIVELY** searches and finds an integer from a given binary search tree (**BST**).
2. Write a method that **RECURSIVELY** prints all the elements of a binary search tree in a sorted order.
3. An adjacency matrix is given below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** | **F** | **G** |
| **A** | **0** | **1** | **0** | **1** | **1** | **0** | **0** |
| **B** | **0** | **0** | **0** | **0** | **0** | **0** | **1** |
| **C** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |
| **D** | **0** | **0** | **1** | **0** | **0** | **0** | **1** |
| **E** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **F** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| **G** | **0** | **0** | **0** | **0** | **1** | **1** | **0** |

1. Draw the equivalent graph.
2. Simulate BFS algorithm on the output of a.
3. Simulate DFS algorithm on the output of a.