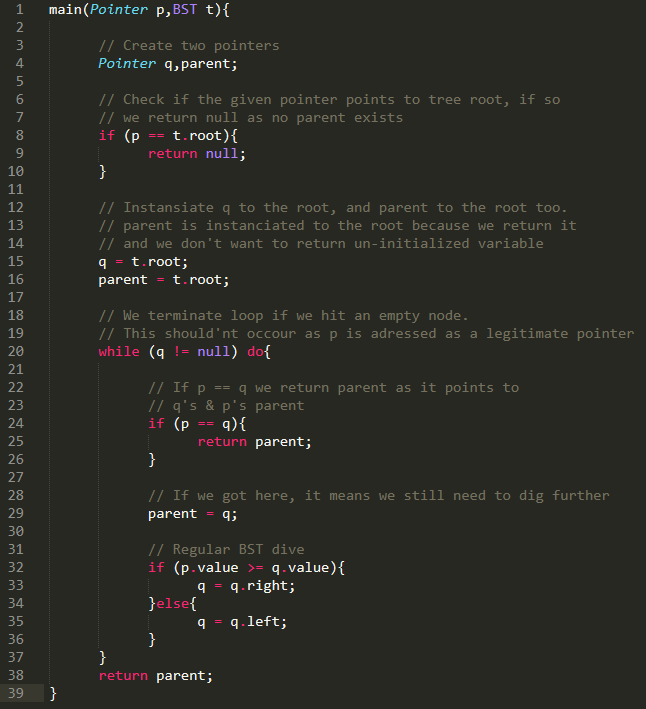
|  |
| --- |
| DATA STRUCTURS, Ex06 |
| Aviad Hahami302188347 |

# Q1.

\*note – This algorithm is correct assuming 2 nodes with equal value will be treated as “bigger than” (i.e. if we have two “8”, then one will be the sub-tree root, and the other will be the right child)



# Q7.

Let person X be as follows:

Description: We will use an augmented interval tree as described by Cormen.

Each node will contain a person X, the person’s lifespan as the interval and the maximum high value among the tree. Below there’s an example of a node in the tree:

X (Person data)

Life span = [X.death, X.birth]

Max high value = Maximum value of birth in sub-tree

|  |  |
| --- | --- |
| Insert a person X into the data structure | 1. Create a new tree node as formatted above. 2. Insert to tree according to regular interval tree properties |
| Remove a person X from the data structure. | 1. Find person X in tree 2. Remove from tree according to regular interval tree properties |
| Given a new person, X, find at least one name of a person Y that lived in the same period. | 1. Generate a “life span” for X. 2. Search the tree for a corresponding interval according to regular interval tree properties. |
| Given a person, X, return another person, Y, that was born in the same month and year as X, if one exits. |  |
| Give the name of the person that was born first |  |
| Give the name of the person that passed away last | 1. Return name of the person in the most-right node. |