Université d'Ottawa
Faculté de génie

École d'ingénierie et de
technologie de l'information

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Information
Technology and Engineering

# Assignment 5 (50 points, 5%)

CSI2110

**Due date is Dec 6, but you may submit to virtual campus by Dec 11, 23h55 with no late penalty.**

***Question 1.*** **[10 points]**

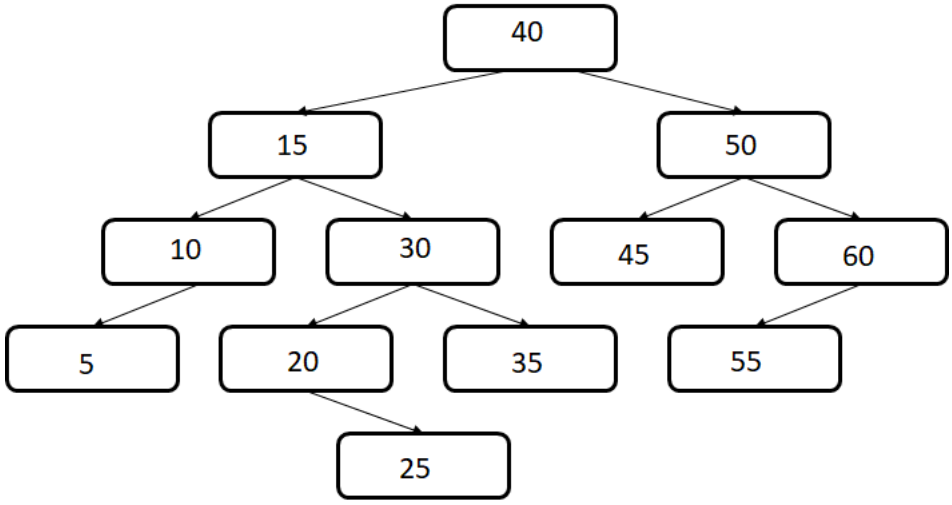
***a)*** Insert the following keys into a **binary search tree**

56, 37, 86, 24, 32.

i. Show the resulting tree and indicate the balancing factor in each node of this tree.

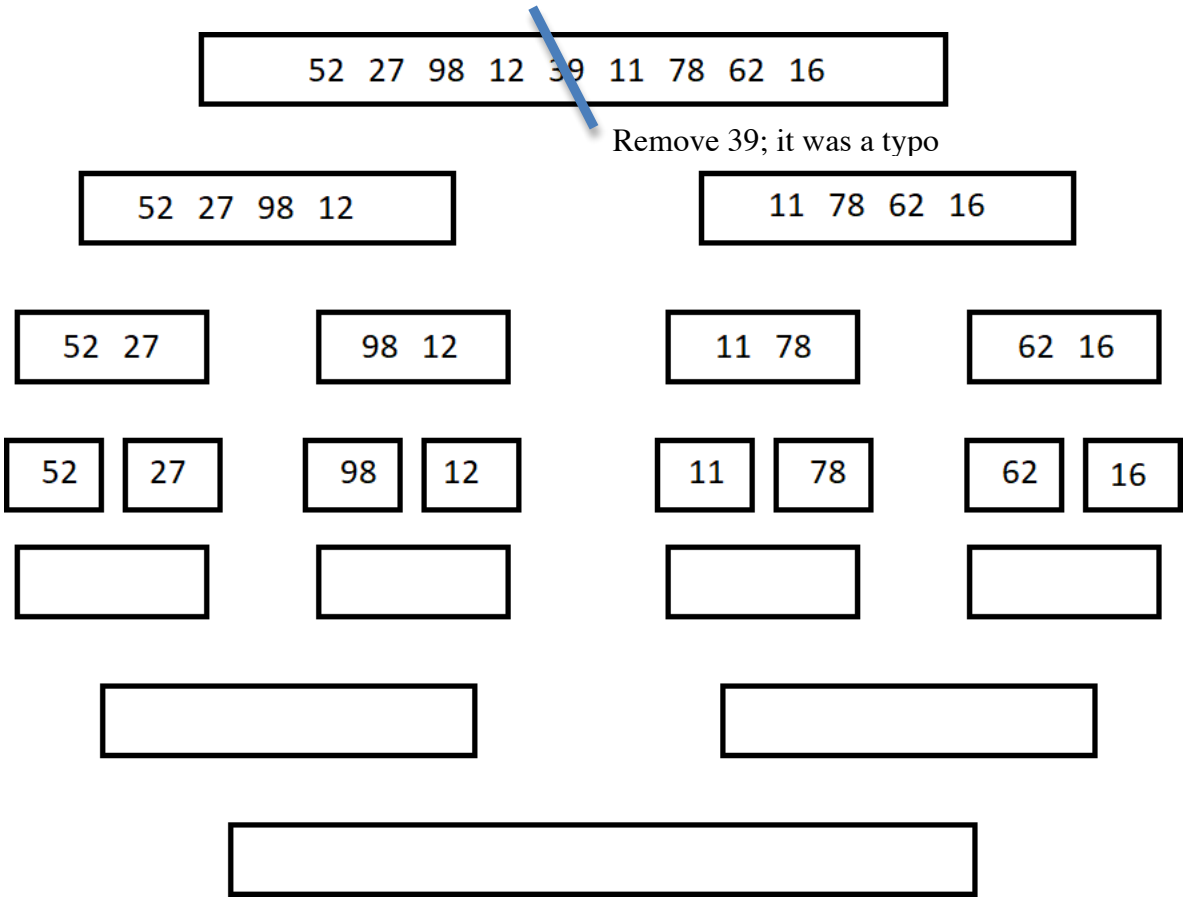
ii. Now, suppose this is an AVL tree. After the last insertion, apply the rebalancing algorithm in order to balance this tree, and show the resulting tree.

b) Remove key 45 from the following AVL tree and show the resulting tree (note that dummy leaves have been ommitted from the drawing but you should consider in your simulations). Display the tree after each rebalancing operation.



***Question 2.*** **[10 points]**

a) The drawing below illustrates the recursive process of the mergesort algorithm. The top half of this graph shows the subdivision of all sub-arrays generated by the recusive calls. The second half shows all the merge operations performed. Complete this graph by showing the content of each subarrays resulting from these merge operations

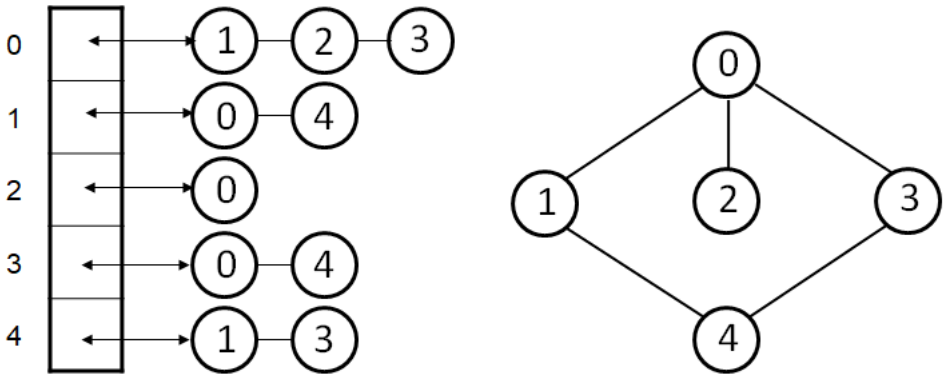



b) The quicksort algorithm can also be illustrated by a similar drawing. The top part of the drawing would be the result of all partition operations that subdivide each sub-arrays into 2 sub-arrays until each of them contain only one element. The second part would simply show the combination of resulting two arrays into one.
There is however one important difference with the previous sort. Mergesort always split one array into two arrays of same size. With quicksort the partitioning most often result into sub-arrays of different sizes such that the call tree will probably be unbalanced.

Illustrate the recursive calls of Quicksort using the array above and by always using the left-most element of a sub-array as pivot when performing partitions.

***Question 3.*** **[10 points]**

Consider a graph with the given representation using adjacency lists.




a) Traverse the graph using depth-first search (DFS) from node 0.
List all nodes in order of traversal.
List all the discovery edges and all the back edges, displaying the edges in the direction they receive their labels. (For example, if you visit vertex **a** then you use edge **{a,b}** to discover vertex **b**, then **(a,b)** is a discovery edge rather than **(b,a).**

b) Traverse the graph using breath-first search (BFS) from node 0.
List all nodes in order of traversal.
List all the discovery edges and all the cross edges, displaying the edges in the direction they receive their labels.

c) Traverse the graph using depth-first search (DFS) from node 4.
List all nodes in order of traversal.
List all the discovery edges and all the back edges, displaying the edges in the direction they receive their labels.

***Question 4.*** **[10 points]**

We will now use graphs in order to find the right persons to contact in order to meet a given person. For this, we will use the concept of degree of separation. If a person knows another person, then they have a degree of separation equals to 1. If a person X knows a person Y that knows Z, but X doesn't know Z, then X and Z have a degree of separation of 2.

Next we give a table showing degrees of separation between a group of people.
.

| | Jean | Paul | Sam | Kim | Emma | Joe | Jack | Jane | Ali | Bob | Luce | Wen |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Jean | | 1 | | | | 1 | | 2 | | 2 | 2 | |
| Paul | 1 | | 1 | | | | | | | | | |
| Sam | | 1 | | 3 | | | 1 | | | | | |
| Kim | | | 3 | | 1 | | | | | | | |
| Emma | | | | 1 | | | | | 1 | | | 2 |
| Joe | 1 | | | | | | 2 | | | | | |
| Jack | | | 1 | | | 2 | | | 2 | | | |
| Jane | 2 | | | | | | | | 3 | 1 | | |
| Ali | | | | | 1 | | 2 | 3 | | | 2 | 2 |
| Bob | 2 | | | | | | | 1 | | | | |
| Luce | 2 | | | | | | | | 2 | | | |
| Wen | | | | | 2 | | | | 2 | | | |

Jean wants to meet Emma. What is the chain of persons to meet that will result in the lowest possible degree of separation with Emma? Explain how to use the Dijkstra algorithm to solve this problem. Show each of the steps that will be executed.

***Question 5.*** **[10 points]**

A hash table uses an array of size 17 with a has function h(k)= k mod 17 with linear probing. Show the array after the insertion of the following keys

66 20 50 75 104 100 17 24 70 25