Week 10 - Graphs

Daniel Alyoshin

CSCA48 - TUT002

March 19, 2026

Content Covered so Far

- C programming
- Memory model
- Arrays and strings
- Pointers
- CDTs & ADTs
- Linked lists
- Testing & Debugging
- Binary Trees & BSTs
- Big O & Complexity Analysis
- Recursion
- Intro to Graphs

Brainstorming

Quickly recall: What is an **adjacency list**?

Brainstorming

Quickly recall: What is an **adjacency list**?

Idea: What if we wanted to use an **adjacency BST** instead...?

Brainstorming

Quickly recall: What is an **adjacency list**?

Idea: What if we wanted to use an **adjacency BST** instead...?

What would that looks like?

Brainstorming

Quickly recall: What is an **adjacency list**?

Idea: What if we wanted to use an **adjacency BST** instead...?

What would that looks like?

- Array with N entries each representing one of our N nodes.

Brainstorming

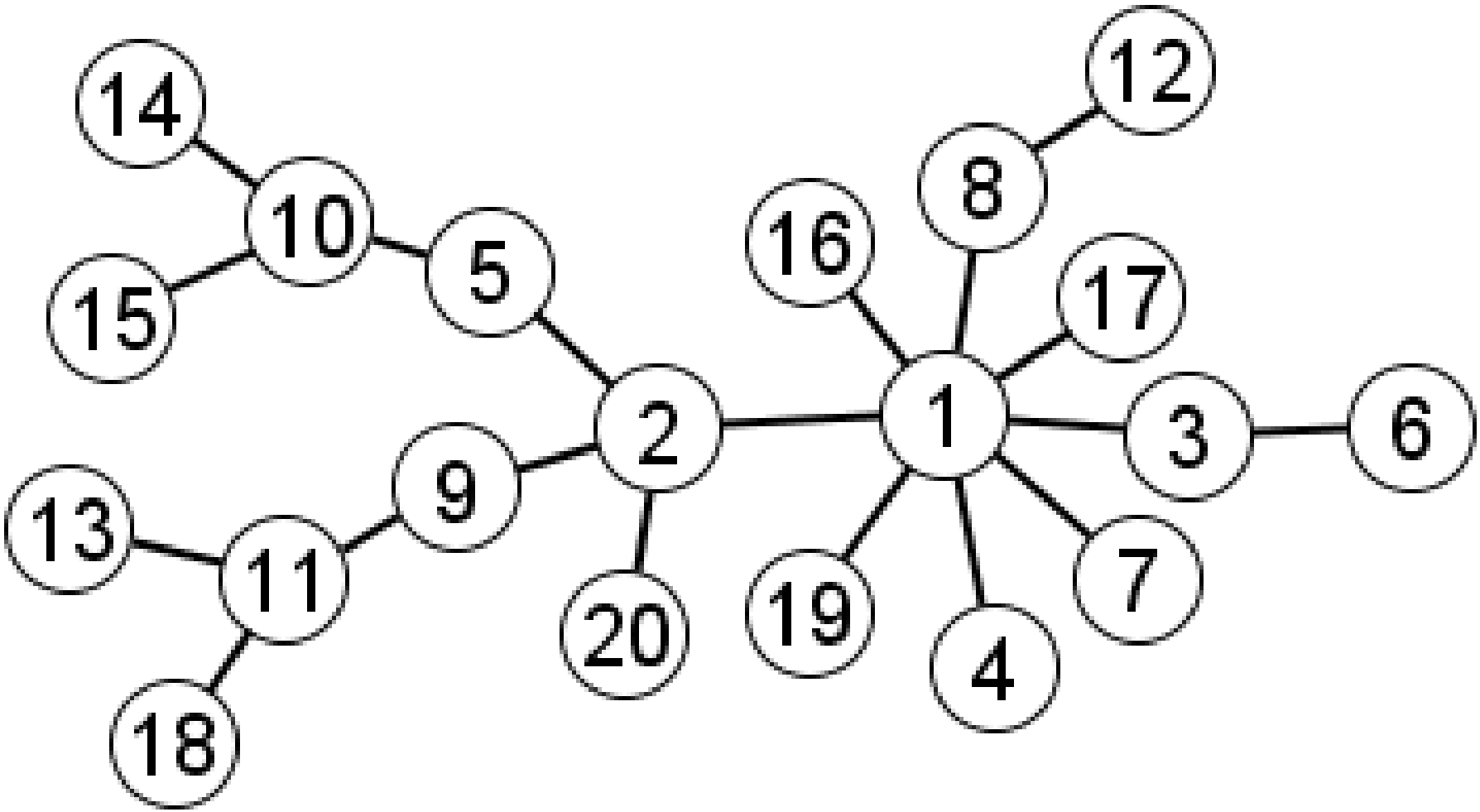
Quickly recall: What is an **adjacency list**?

Idea: What if we wanted to use an **adjacency BST** instead...?

What would that looks like?

- Array with N entries each representing one of our N nodes.
- i^{th} element of the array contains a pointer to the root of the adjacency BST storing the edge information for the i^{th} node in the graph.

Your First Task



Your Second Task

For our adjacency BST, figure out the **average** case Big O complexity of the following:

- Adding an edge
- Removing an edge
- Adding a new node (assume the array is full)
- Removing a node
- Edge query

Your Second Task

For our adjacency BST, figure out the **average** case Big O complexity of the following:

- Adding an edge -

Your Second Task

For our adjacency BST, figure out the **average** case Big O complexity of the following:

- Adding an edge - $O(logN)$
- Removing an edge -

Your Second Task

For our adjacency BST, figure out the **average** case Big O complexity of the following:

- Adding an edge - $O(logN)$
- Removing an edge - $O(logN)$
- Adding a new node (assume the array is full) -

Your Second Task

For our adjacency BST, figure out the **average** case Big O complexity of the following:

- Adding an edge - $O(logN)$
- Removing an edge - $O(logN)$
- Adding a new node (assume the array is full) - $O(N)$
- Removing a node -

Your Second Task

For our adjacency BST, figure out the **average** case Big O complexity of the following:

- Adding an edge - $O(logN)$
- Removing an edge - $O(logN)$
- Adding a new node (assume the array is full) - $O(N)$
- Removing a node - $O(N \cdot logN)$
- Edge query -

Your Second Task

For our adjacency BST, figure out the **average** case Big O complexity of the following:

- Adding an edge - $O(logN)$
- Removing an edge - $O(logN)$
- Adding a new node (assume the array is full) - $O(N)$
- Removing a node - $O(N \cdot logN)$
- Edge query - $O(logN)$

Discussion

When should we opt to use an adjacency BST vs. an adjacency list?

Think about the following:

1. What information do we need to consider in making this decision?
2. What advantages/disadvantages exist for each of our options?
3. Does it matter if the graph is directed or undirected?

Reminders

- Each student to submit their own photo or scan of your work which can be the same as that submitted by members of their group.
- Ensure your file is named **exactly** as instructed on Quercus.
- You are **not** supposed to keep working on this, you will not be marked on correctness so as long as you submit work that shows you were engaged during this tutorial you will get full marks.