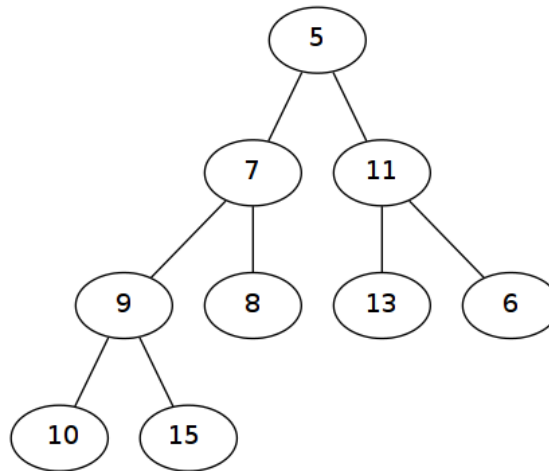


Example Exam Questions

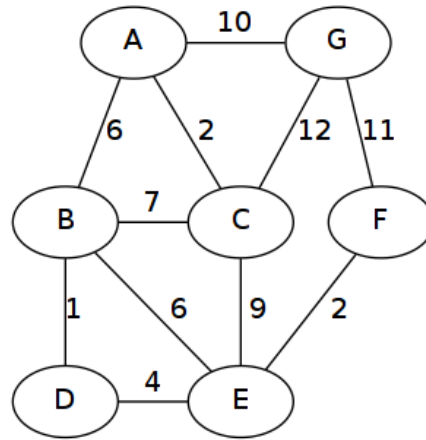
MATH 8650

Timo Heister, heister@clemsn.edu

1. Draw the graph with edges AB, BC, CD, DE, DF, CG as a tree rooted in C and list all vertices in breadth-first order (use alphabetical ordering if you have a choice). Explain an algorithm to perform a breadth-first search. What data structure would you use?
2. We consider Heaps/Priority Queues implemented using a complete binary tree. Note that we consider min-heaps here, meaning the element with highest priority compares “smaller” than every other element.
 - (a) Define the heap property for a given binary tree.
 - (b) Is the following tree a heap? If not, rearrange the minimum number of nodes to achieve this.



- (c) What does the `pop()` operation of a heap do and return (in general)?
 - (d) Execute two `pop()` operations on the heap in (b), annotate what you do, and draw intermediate steps as necessary. Include the final heap in form of a tree.
 - (e) State the algorithmic complexity of `pop()` (with respect to the number of elements n) and give a short argument for the worst and best case.
3. Let G be a weighted, connected Graph.
 - (a) Define what a minimum spanning tree (MST) of G is.
 - (b) Find the MST of the following graph using Kruskal’s algorithm. Number and annotate each step of the algorithm and draw the intermediate graphs with “colors” after you change a color. Also give the final answer as a list of edges.



- (c) Is the solution you found unique, why or why not? Can you say anything about uniqueness without executing the algorithm in (b) just by looking at the graph G ?
4. (a) Sketch the memory layout of a deque how it looks like in CPython.
- (b) In c++ the std::deque can only store objects of a single type T (in contrast to Python), why does this give better performance and memory savings?
- (c) You are tasked to find the minimum value in a list of numbers (stored in an array). How would one parallelize this search with multiple threads and do you need any locks/atomic variables?

No

