This homework covers graphs, graph traversal via BFS, and some BFS-based graph algorithm. It also revisits data structure design and some applications of specialized data structures and fundamental algorithms. It is due in class Wednesday, March 4.

See the Policies page on the course website for information about revise-and-resubmit, late work, and academic integrity as it applies to homework.

Write your solutions carefully — your work should be neat, readable, and organized.

Review homework 3 for hints and tips about solving and writing up design problems. In addition, when building on a known solution (data structure or algorithm), you do not need to repeat the details of the known thing — just address how your additions or modifications fit into that structure. For example, BFS has placeholders for process vertex early*,* process edge*, and* process vertex late *steps so you can refer to those pieces. The shortest unweighted path algorithm covered in class is thus just BFS where* `dist[`v`]` *is initialized to 0 for the start vertex and* ∞ *for all others and the "process edge (u,v)" step is* `dist[`v`] = dist[`u`]+1`.

---

1. Do the homework #5 drill problems on Canvas. (Look for `hw5 drill` in the Quizzes section.)

2. ADM 7-6, page 236. Take the section header "Traversal" as a hint for the solution strategy — for the $O(n + m)$ algorithm, consider how traversing the graph (and specifically traversing the graph using BFS, since that is all that has been discussed in class as of 2/25) can be used to solve the problem. To reduce the running time to $O(n)$, note that in this case you are not being asked to come up with a new algorithm. Instead, better analyze the running time of your BFS-based algorithm — why will it not have to traverse more than $O(n)$ edges?

3. ADM 3-20, page 104.

4. ADM 3-24, page 105. Take the section header "Applications of Tree Structures" as a hint for the solution strategy — consider a (binary) tree at the ADT level and design your algorithm in terms of insert, remove, and find operations on the tree.