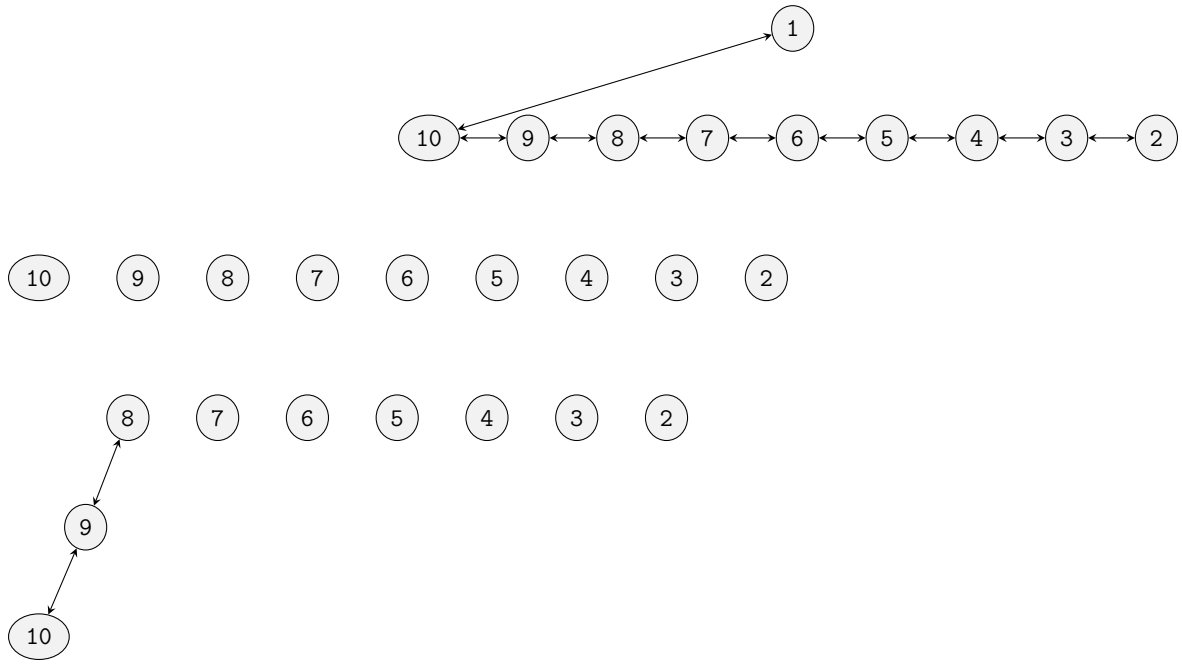
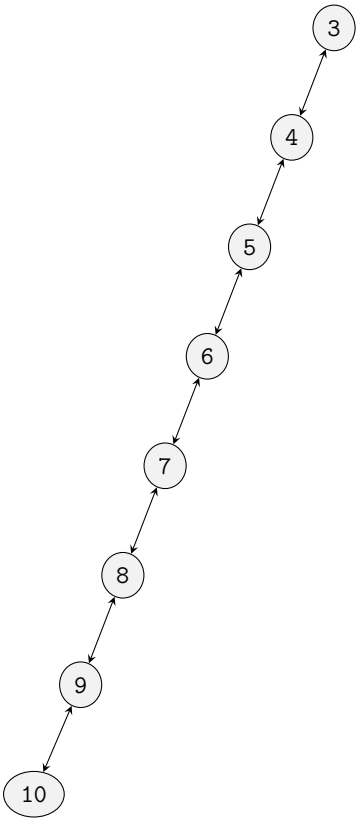
Today: Pairing heap; § 23.2.
Next class: HW04 part 2 due. Graphs, shortest paths; §§ 14.*, 23.2.3,
Reminders: Term project plans. Read material, incl. code, *before and after* class.

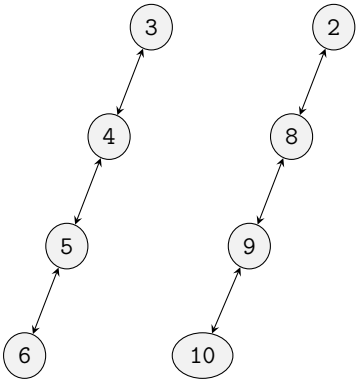
1. Write your group identifier (e.g., C3) and its members' names below. Underline your name.

2. Consider an initially empty structure similar to a pairing heap, but that is maintained using a simple one-pass linking strategy in which subtrees are merged one at a time in left-to-right order.
   (a) Trace the insertion of the keys $1, 2, \ldots, 10$ into this heap, depicting the intermediate trees after 2 and 5 insertions.
   (b) Explicitly depict the null nodes in the tree depicted below.
   (c) Use dashed lines to depict the *abstract tree* corresponding to this concrete tree.
   (d) Then trace two *deleteMin* operations.
   (e) Then trace one *decreaseKey* operation that changes the key 7 to 2.



[additional space for answering the earlier question]





3. Repeat Question 2 using a two-pass linking strategy that merges pairs of subtrees left to right in the first pass and then merges the merged pairs also in left-to-right order in the second pass. (In the second pass, we proceed left-to-right, merging the result of the previous merges in this pass with the next subtree.)

4. Repeat Question 3 using a right-to-left second pass. Explain any differences between this strategy and that of the textbook.