EC 504
Spring, 2021
HW 3

Due Monday, Feb. 28, 8PM on Gradescope.

1. (15 pts) Draw the Fibonacci heaps that result from the following sequence of operations. Don't bother drawing the circular linked lists at each level of the trees, only at the root level.

   - insert 1,2,3,4,5,6,7,8,9,10,11,12.
   - then delete 1 (extract min);
   - then change 8 to 0 (reduce key);
   - then change 7 to 1.

   Solution: Note that there are multiple solutions, all of which are equivalent, because of the choice of how we merge trees when we delete 1. What is important is that the initial result after deleting 1 has the right shape tree and has the right tree structure in terms of binomial trees. The pictures below illustrate one such solution. The pictures show the heap after insert all, after delete 1, after changing 8 to 0, and at the end. The green node is "marked".

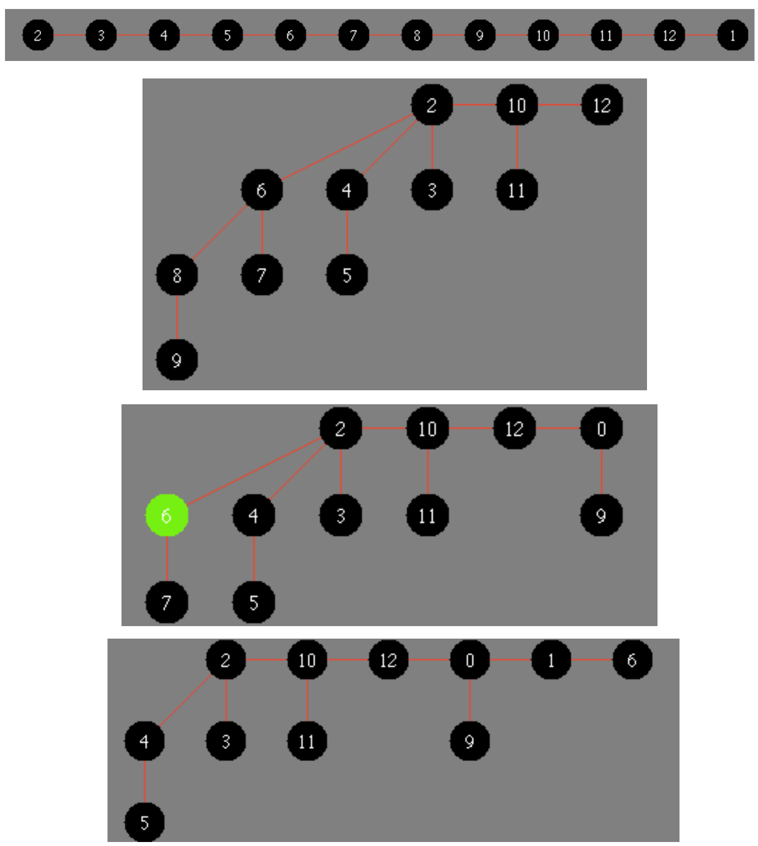


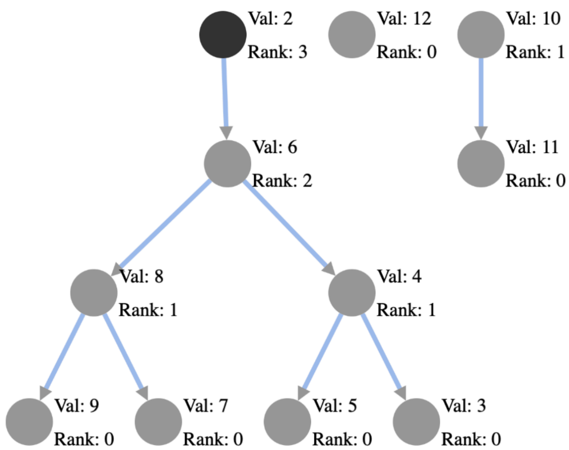
Figure 1: Solution to Problem 4.c

2. (15 pts) Draw the rank-pairing heaps that result from the following sequence of operations.

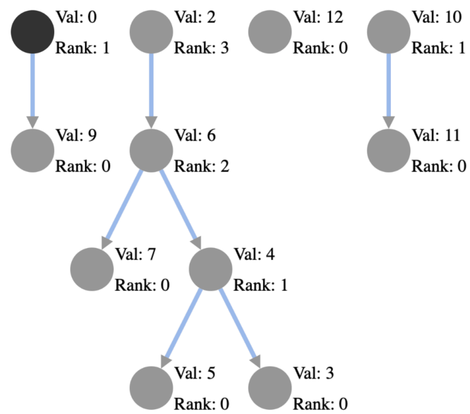
   - insert 1,2,3,4,5,6,7,8,9,10,11,12.

- then delete 1 (extract min) with recursive merging;
- then change 8 to 0 (reduce key);
- then change 7 to 1.
- then extract min (delete 0) with recursive merging;

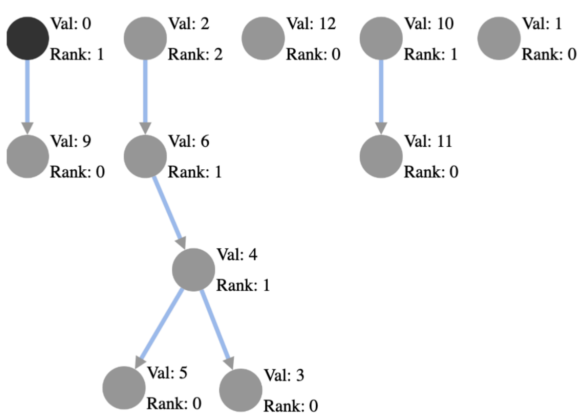
Solution: We don't show the straight line of nodes after 12 inserts. The solutions below include explanations.



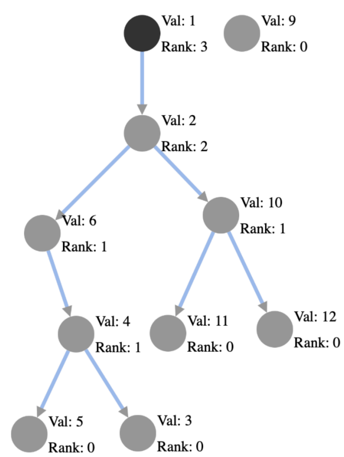
Solution to Problem 2(a). Note that there are multiple solutions, all of which are equivalent, because of the choice of how we merge trees when we delete 1. The trees below show one possible solution. What is important is that the shape of the trees is the same after delete 1 with recursive merging, and the ranks are correct at each level, with the trees satisfying heap order to the left.



Solution to Problem 2(b), reduce 8 to 0. Wherever 8 is in the heap, 8 and its left subtree become a new tree, with the key of 8 reduced to 0; the node where 8 was is replaced by 8's right child. Ranks need to get adjusted appropriately.

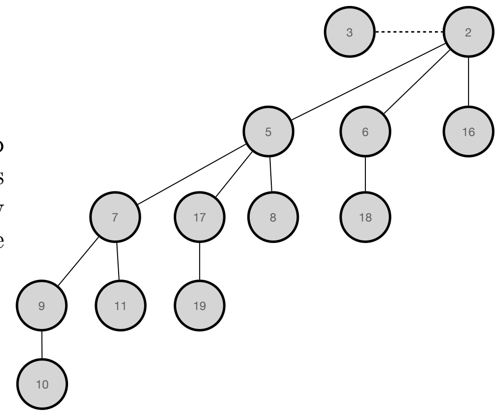


Solution to Problem 2(c), reduce 7 to 1. Wherever 7 is in the heap, it and its left subtree become a new tree, with the key of 7 reduced to 1; the node where 7 was is replaced by 7's former right child. Ranks need to get adjusted appropriately.

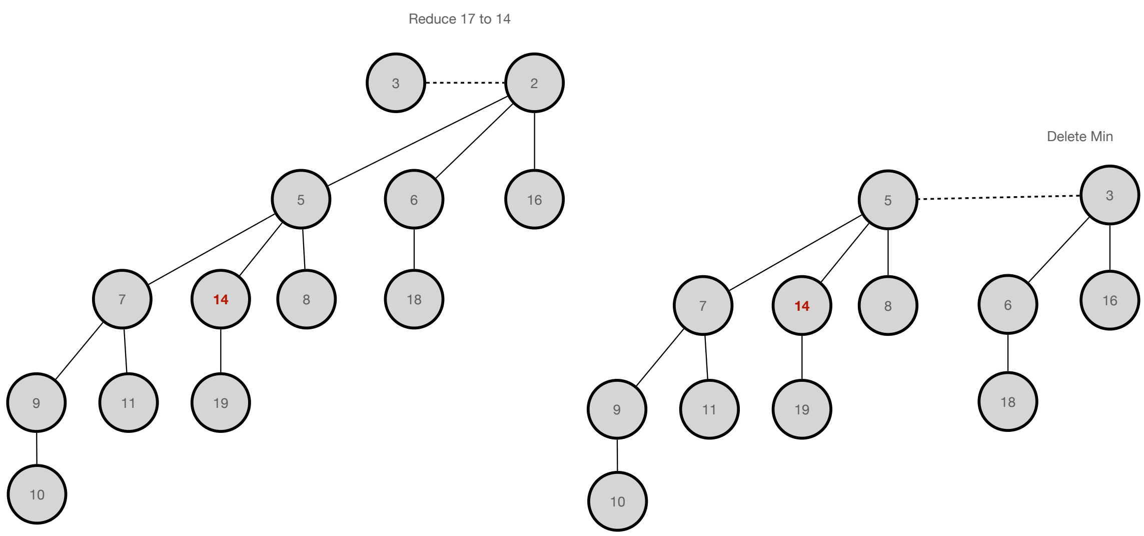


Solution to Problem 2(d), extract min. The important aspect is the shape of the trees and the ranks. The values will be in different locations depending on the merge order chose.

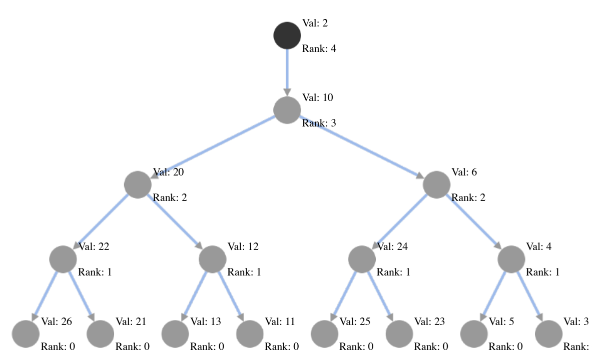
3. (8 pts) Old midterm question: Consider the Fibonacci min-heap illustrated on the right. Show the Fibonacci heap that results after the following operation: Reduce key 17 to 14. Then, show the Fibonacci heap that results when you delete minimum in the resulting heap from the first part.



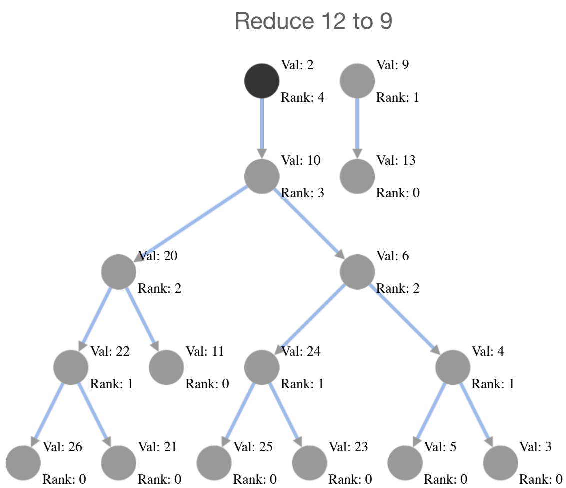
Solution: Here is the solution. The first decrease key changes nothing because it still has the heap property.

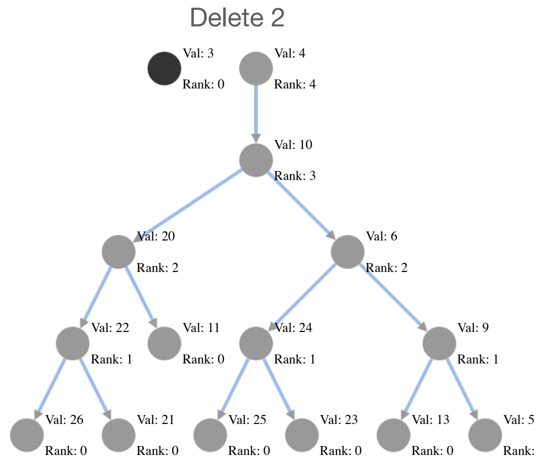


4. (8 pts) Old midterm question: Consider the rank-pairing min-heap illustrated on the right. Show the rank-pairing heap, including ranks, that results after the following operation: Reduce key 12 to 9. Then, show the rank-pairing heap that results when you delete minimum in the resulting heap from the first part, using recursive merging, showing the resulting ranks.



Solution:





5. (15 points) For the KMP algorithm, Compute the prefix function for the following string: "abracadabra".

   Solution:

   $\pi = [0, 0, 0, 1, 0, 1, 0, 1, 2, 3, 4]$

6. (15 points) For the Aho-Corasick algorithm, compute the suffix trie that would be used to search for the following strings simultaneously:

```
"the"
"other"
"otter"
"often"
"threw"
"these"
"rocks"
```

   Solution: See figure below

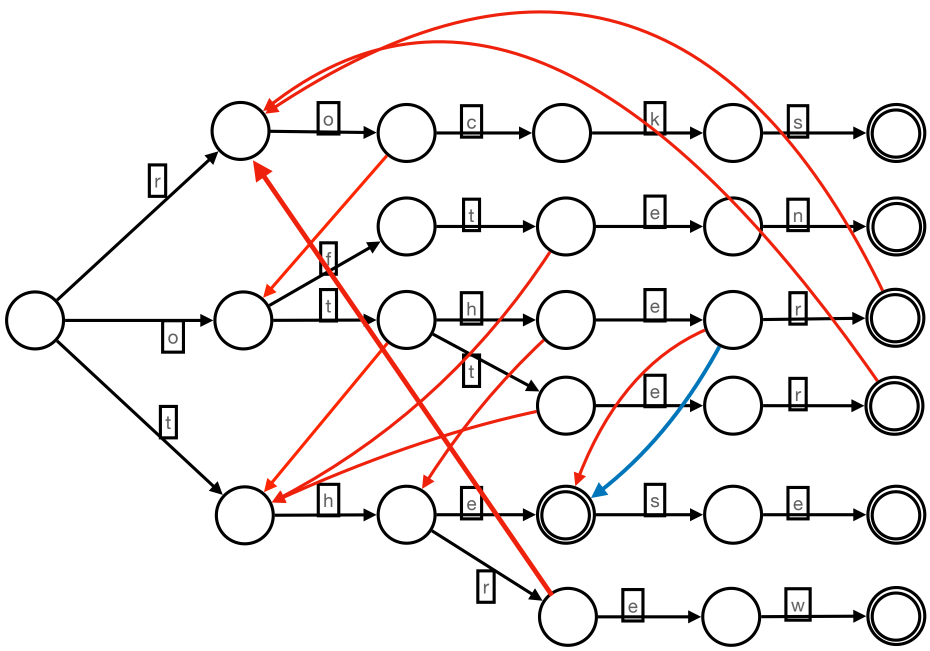


Figure 2: Solution for Problem 6. Suffix links to root not shown.