Quiz/Homework Assignment #3
CSE326
Homework due date:
Wednesday, January 26th, beginning of class

January 20, 2000

This quiz is homework assignment #3. Any question you answered correctly in class are considered correct for the purposes of the assignment. Any question you did not answer or which you answered incorrectly is now a short answer question for the homework assignment due in class on Wednesday. We will hand back the graded quiz on Friday.

In *italics* are instructions on how to answer each question for homework. The correct letter answer is listed after each question.

Good luck!

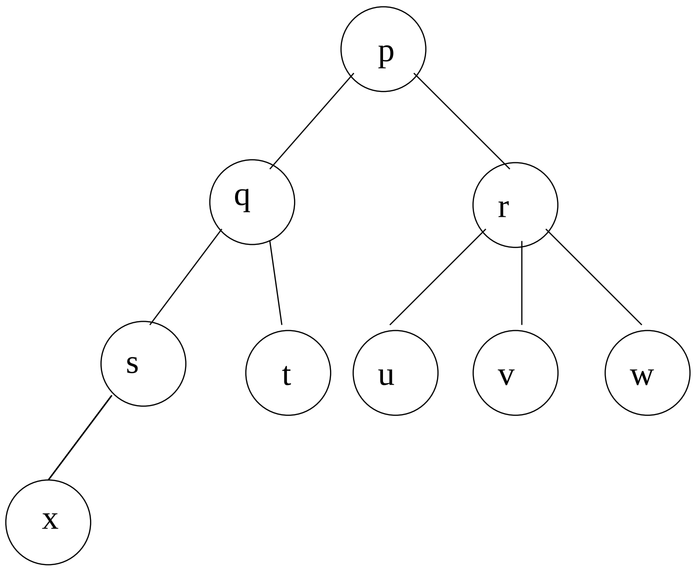

Figure 1: The tree from quiz #2. Use this tree to answer questions 1-2. Note that the priorities of the nodes are not necessarily equal to their letter labels!

1. *Explain why one of* **u**, **v**, *or* **w** *must be removed. Explain why* **x** *need not be removed. Describe the heap-order property, and explain why it may not be obeyed in this tree.*

   Which of the following must be done to ensure that this tree is a binary heap?

   (a) Remove **x**
   (b) Remove one of **u**, **v**, or **w**
   (c) Ensure that the heap-order property is obeyed
   (d) Both (b) and (c)
   (e) All of the above

   Correct answer(s): **(d)**

2. *Explain what happens on a call to deleteMin and why the answer follows from this process. You* may *assume that all nodes have unique priorities.*

   Assume that all necessary changes have been made to ensure this tree is a heap. After one call to deleteMin, which node will be at the root?

   (a) **p**
   (b) **q**
   (c) the lesser of **q** and **r**
   (d) **r**
   (e) **x**

   Correct answer(s): **(c)**

3. *Explain why heap based priority queues* do *need to be able to compare the priority of data items. Explain why they do* not *need to know actual numerical values describing the priorities.*

   Stacks and Queues require almost no knowledge about the data they store other than its size. What knowledge (other than size) do heap based priority queues require?

   (a) None.

   (b) They need to be able to compare the priority of the data items.

   (c) They need to be able generate numerical values describing the priority of the data items.

   (d) They need to know the exact structure of the data items.

   (e) They need coffee, and lots of it.

   Correct answer(s): **(b)**

4. *Analyze the running time of the creation of the heap. Analyze the running time of the printout of the numbers. Give an overall runtime. You may assume that $O(n \log n)$ is a tight upper bound on $\log n!$. If your answer on the quiz was a loose bound, also explain why it is loose.*

   Here's an algorithm to sort n numbers with a heap. Start with an empty heap and insert each number into the heap sequentially. Then, remove a number from the heap and output it, repeating until the heap is empty again. How long does this take? (Tight upper bound, please.)

   (a) $O(\log^2 n)$

   (b) $O(n^2)$

   (c) $O(n)$

   (d) $O(n \log n)$

   (e) $O(n^2 \log n)$

   Correct answer(s): **(d)**

5. *Explain why each of (a), (b), (c), and (d) may be concerns in choosing d in a d-heap.*

   Which of the following is **not** a concern in choosing d in a d-heap?

   (a) the size of cache lines

   (b) making d a power of 2

   (c) the size of elements in the heap

   (d) the relative number of inserts and deleteMins

   (e) the smallest key value in the heap

   Correct answer(s): **(e)**

6. *Draw an example of the smallest and of the largest possible 5-heaps, and note their size. Draw the arrays that would contain those heaps.*

   What is the smallest and largest number of entries in an array that a 5-heap of depth 2 can take (**not** including the element for size)?

   (a) 7 and 31

   (b) 6 and 30

   (c) 25 and 125

   (d) 5 and 25

   (e) None of these; a 5-heap cannot be stored in an array

   Correct answer(s): **(a)**

7. *Describe how deleteMin and insert are actually implemented in leftist and skew heaps. Explain why these are efficient.*

   Which of the following is true about insert, deleteMin, and merge in leftist and skew heaps?

   (a) insert is a merge

   (b) merge is a sequence of inserts

   (c) merge is a sequence of deleteMins

   (d) deleteMin is a sequence of merges

   (e) those heaps don't implement merge

   Correct answer(s): **(a)**

8. *Give the null path length of each node in each tree. Explain why these are therefore leftist heaps.*

   In order to make these into leftist heaps, what needs to be done?

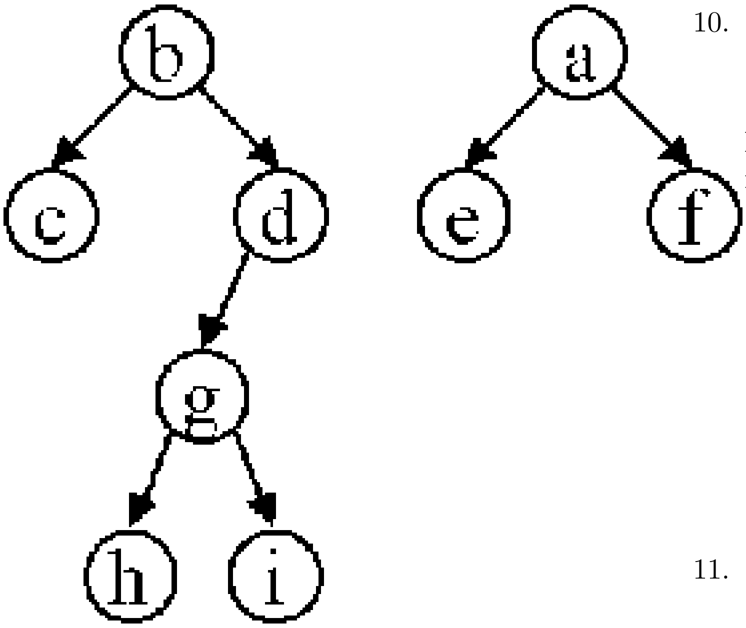

Figure 2: Two binary trees. Use this tree to answer questions 8-10. This time, assume that the priorities of the nodes **are** equal to their letter labels.

(a) Swap **b**'s right and left subtrees

(b) Put **f** under **e**

(c) Put **g** under **c**

(d) (b) and (c)

(e) Nothing; these are leftist heaps.

Correct answer(s): **(e)**

9. *Draw out each step of the merge process in the same manner that we did in class (use recursive not iterative merge).*

   If we merge these as leftist heaps, which of the following is true of the resulting tree?

   (a) Every node has a null path length of 1.

   (b) Every node has zero or two children.

   (c) The left path is: **a** $\Rightarrow$ **b** $\Rightarrow$ **d** $\Rightarrow$ **f**.

   (d) Every node has a null path length of 0.

   (e) None of these.

   Correct answer(s): **(b)**

10. *Draw out each step of the merge process in the same manner that we did in class (use recursive not iterative merge).*

    If we merge these as skew heaps, which of the following is true of the resulting tree?

    (a) Every node has a null path length of 1.

    (b) It is a leftist tree.

    (c) The left path is: **a** $\Rightarrow$ **b** $\Rightarrow$ **d** $\Rightarrow$ **f**.

    (d) One node has a null path length of 2.

    (e) None of these.

    Correct answer(s): **(c)**

11. *Note the clarification in this question (on part (a)). Explain why (c) and (d) are not conditions which favor skew heaps over d-heaps. Explain why (a), (b), and (e) together constitute a good situation to use skew heaps over d-heaps.*

    Which of the following together might constitute a good situation to use skew heaps over d-heaps? (Bubble in all that apply!)

    (a) Using minimal memory at any given point in time without sacrificing asymptotic speed performance is vital.

    (b) The size of the heap fluctuates wildly.

    (c) The size of the heap is stable and known.

    (d) Speed of merges is not essential.

    (e) Speed of each individual deleteMin is not essential.

    Correct answer(s): **(a, b, e)**