YORK UNIVERSITY

2009 WINTER TERM EXAMINATION

CSE4101 / 5101
Advanced Data Structures
Professor Ruppert

Duration: 3 hours

Examination Aids: One sheet of paper measuring $8\frac{1}{2} \times 11$ inches, handwritten on both sides.

- There should be 10 pages in the exam, including this page.
- Write all answers on the examination paper. If your answer does not fit in the space provided, you can continue your answer on the back of a page or on page 10, indicating clearly that you have done so.
- Write legibly.

Name ______________________________

Student Number ______________________________

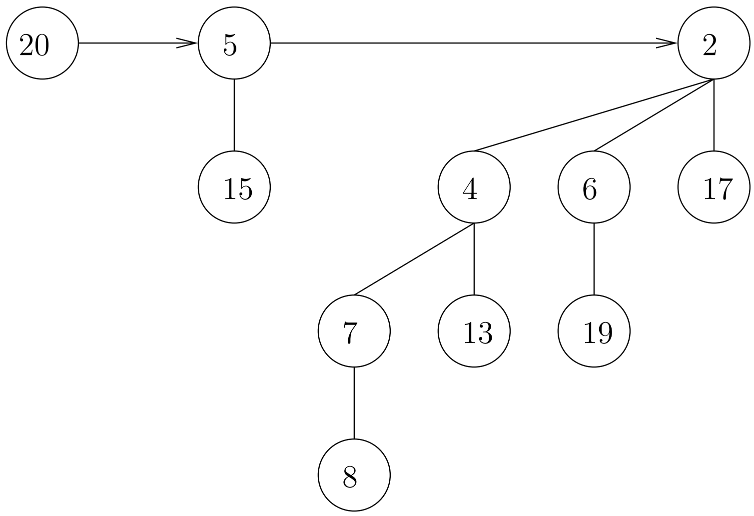
1.	____________	/4
2.	____________	/3
3.	____________	/2
4.	____________	/3
5.	____________	/3
6.	____________	/4
7.	____________	/4
8.	____________	/5
9.	____________	/6
10.	____________	/11
11.	____________	/5
Total:	____________	/50

1. [4 marks] A sequence of stack operations is performed on a stack whose size never exceeds k. After every k operations, a copy of the entire stack is made for backup purposes. What is the worst-case total time for a sequence of m stack operations (including the time to make the backup copies)? *Briefly* justify your answer.

2. [3 marks] Can the black heights of nodes in a red-black tree be maintained as fields in the nodes of the tree without affecting the asymptotic performance of any of the red-black tree operations? *Briefly* justify your answer.

3. [2 marks] In class, we showed that the set of *all* functions from $U = \{0, \ldots, N-1\}$ to $\{0, \ldots, m-1\}$ is a universal class of hash functions. Why would it be a bad idea to choose a random element of this class as a hash function when implementing perfect hashing?

4. [3 marks] Show what the following binomial heap would look like after an EXTRACT-MIN operation has been performed.



5. [3 marks] What is the largest possible number of red nodes in a red-black tree that stores 10 keys? Draw a red-black tree with 10 keys that has that many red nodes, indicating the colour of each node.

6. [4 marks] If a B-tree contains k keys and n nodes, define its potential to be $k - n$. Use this potential function to give a good upper bound on the total number of node splits that occur during a sequence of m insertions into an initially empty B-tree.

(Note: I am not asking about the running time, but only about the number of node splits.)

7. [4 marks] Suppose you are given an unsorted linked list of n integers drawn from the set $U = \{0, \ldots, N-1\}$. Give a randomized algorithm that removes duplicates from the list. Your algorithm should have expected running time $O(n)$ and use $O(n)$ space. (You may assume you have a big prime number.)

8. [5 marks] Willemina is given a sequence of n items. The ith item weighs w_i grams. Willemina wants to pack the items into bins. Each bin can hold up to 10 kilograms. Willemina wants to use as few bins as possible. She decides to use the following heuristic (called "best-fit"): she goes through the items in the order they are given to her and puts each one into the bin with the smallest remaining capacity that the item will fit into. In other words, if the item weighs w_i grams, she puts it into the bin whose current contents are closest to $10000 - w_i$ grams, without exceeding that value. (If there is a tie, Willemina can break the tie in an arbitrary way.)

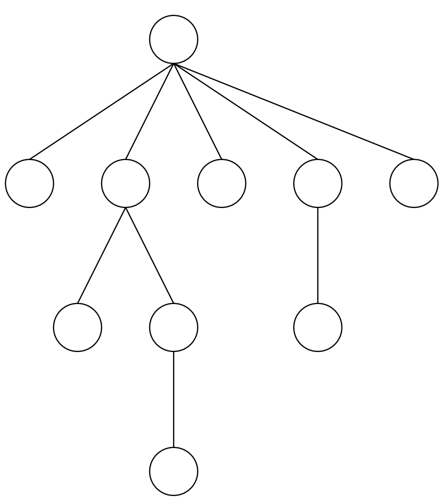
[4] **(a)** Use a red-black tree to design an efficient algorithm that takes $w_1, \ldots, w_n$ as inputs and tells Willemina which bin to put each item into. (Assume $w_i \leq 10000$ for each i.)

[1] **(b)** What is the running time of your algorithm? State your answer in terms of n using big-O notation.

9. [6 marks] Recall the union-find data structure that uses a forest of trees. Each tree represents one of the disjoint sets. Assume that both path compression and union by rank are being used.

[4] **(a)** Prove the following data structure invariant holds at all times: If a tree's root has rank r, then the tree contains at least 2^r nodes.

[2] **(b)** What are the possible values for the rank of the root of the following tree? Briefly justify your answer.



10. [11 marks] Suppose we modify the algorithms for Fibonacci heaps so that a node is cut away from its parent when it loses its *third* child instead of cutting it when it loses its second child.

Define F_d by the recurrence

$$\begin{aligned} F_0 &= 1 \\ F_1 &= 2 \\ F_2 &= 3 \\ F_d &= 3 + \sum_{i=0}^{d-3} F_i, \text{ for } d \geq 3. \end{aligned}$$

Note: if you cannot answer some parts of this question, you should still try to answer the other parts.

[4] **(a)** Prove that $F_d \geq \left(\frac{4}{3}\right)^d$ for all $d \geq 0$.

Question 9 is continued on the next page.

Question 9 continued from previous page

[3] **(b)** Suppose a node v in the modified Fibonacci heap has d children. Explain why the subtree rooted at v contains at least F_d nodes.

[3] **(c)** Prove that, after a CONSOLIDATE operation, the number of roots in the modified Fibonacci heap is $O(\log n)$, where n is the number of nodes in the Fibonacci heap.

[1] **(d)** What is the total cost of a sequence of m INSERT and EXTRACT-MIN operations on the modified Fibonacci heap, assuming the data structure is initially empty? State your answer in terms of m using big-O notation. (You do *not* have to prove your answer is correct.)

CONT'D...

11. [5 marks] A "steap" is an abstract data type that stores a sequence of integers and supports Push(k), Pop and Extract-Max operations. Push and Pop have the same behaviour as in ordinary stacks. The Extract-Max operation deletes the largest integer in the stack.

Describe how to implement a steap so that each operation has worst-case running time $O(\log n)$, when the steap contains n elements.

Hint: The word "steap" comes from a combination of "stack" and "heap".

This nearly blank page is just for additional workspace, if you need it.