UNIVERSITY OF TORONTO
Faculty of Arts and Science
AUGUST 2015 EXAMINATIONS

CSC263H1Y

Duration - 3 hours
Examination Aids: Written and printed material on paper. Non-programmable calculator.
You need at least 35% on this exam to pass the course.

First Name: Last Name:
Student Number:

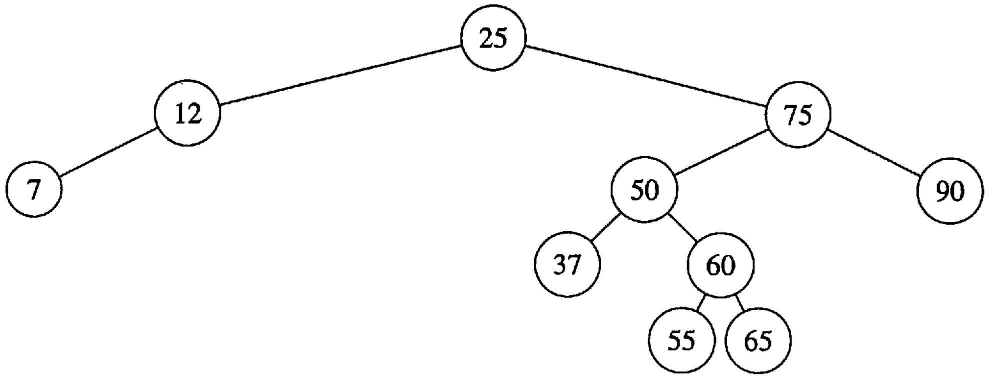
1:	/ 5	10:	/ 5
2:	/ 5	11:	/ 5
3:	/10	12:	/ 10
4:	/ 5	13:	/ 5
5:	/ 5	14:	/ 5
6:	/ 5	15:	/ 5
7:	/10	16:	/ 5
8:	/ 5	17:	/ 10
9:	/10	18:	/ 10
		Total:	/120

1. [5 marks] Prove or disprove: $(\ln n)^2 \in O(\ln(n^2))$.

2. [5 marks] Prove or disprove: $n \lg n \in O(\lfloor n \lg n \rfloor)$.

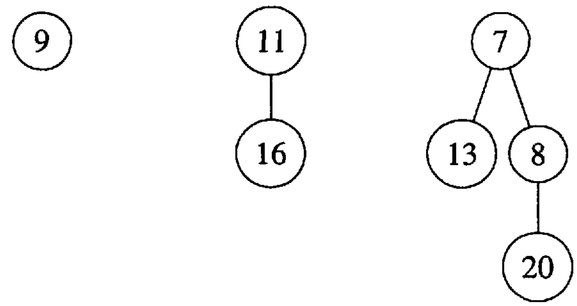
3. [10 marks] Suppose for all natural n, $f(n) \geq 0$, $g_1(n) \geq 0$, and $g_2(n) \geq 0$.
Define $h(n) = \min(g_1(n), g_2(n))$.
Prove or disprove: If $f(n) \in O(g_1(n))$ and $f(n) \in O(g_2(n))$, then $f(n) \in O(h(n))$.

4. [5 marks] Insert 70 into the following binary search tree, then use the weight-balanced tree rebalancing algorithm to rebalance Show the final answer (You may include intermediate states.)

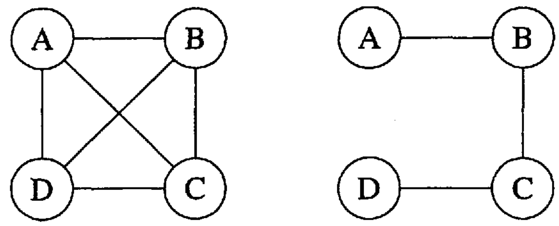


5. [5 marks] Insert the following priorities into a binary max-heap. Show the final answer in tree form. (You may include intermediate states.)
29, 3, 28, 25, 15, 23, 12

6. [5 marks] Start with the following Fibonacci min-heap. Do an extract-min and consolidate. Show the final picture. (You may include intermediate states.)



7. [10 marks] The picture on the left is an undirected graph. The picture on the right is a spanning tree of that graph.



(a) [6 marks] Prove that the spanning tree cannot be produced by breadth-first search.

(b) [4 marks] Assign weights to the 6 edges such that the spanning tree is a mininum spanning tree.

8. [5 marks] Draw the final trees resulting from the following union-find operations. Use union by rank and path compression.

   - The elements are the natural numbers from 1 to 8. Do a makeSet for each element.
   - union(6,4), union(3,6), union(1,8), union(2,5), union(1,2), union(7,6)
   - find(5)

   When unioning two sets of the same rank, merge the second set into the first set, e.g., after union(6,4), 4's parent is 6.

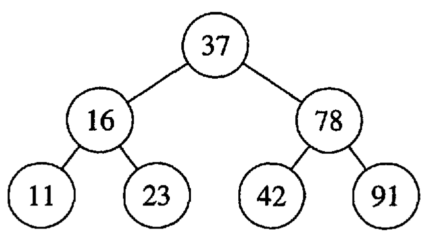
9. [10 marks] My friend Daniel uses a weight-balanced binary search tree and a linked list to implement a set. The operations are coded up this way:

   - insert(k): Prepend k to the list.
   - lookup(k): For each key in the list, insert into the tree. Clear the list. Look for k in the tree.
   - (remove(k) omitted for simplicity.)

   Prove that the amortized time of each operation is in $O(\lg n)$ (if there are n keys).

10. [5 marks] Let A, B, C be events. Suppose $\Pr(C) \neq 0$ and $\Pr(B \cap C) \neq 0$.
Prove $\Pr(A \cap B \mid C) = \Pr(A \mid B \cap C) \Pr(B \mid C)$.

11. [5 marks] Below on the left is a binary search tree. Suppose a lookup is done, and the key to look up is from the table on the right with the given probabilities:



key	probability
16	0.3
23	0.2
37	0.4
40	0.1

What is the expected value of the number of comparisons? Show important steps.

12. [10 marks] Let positive integer n be given. Of the following randomized algorithm, what is the expected value of the final value of t? Show your steps and/or justification.

```
p := 0
t := 0
while p < n do:
    t := t + 1
    c := 0
    for i from 1 to 4
        if random(1, 2) = 1 then:
            c := c + 1
    if c = 3 then:
        p := p + 1
```

13. [5 marks] Insert the following keys in the given order into a hash table of 7 slots. Start from empty. Show the final array.

    - The hash function is $h(k) = k \bmod 7$.
    - Use linear probing for collision handling.

    The keys: 7, 9, 10, 3, 2

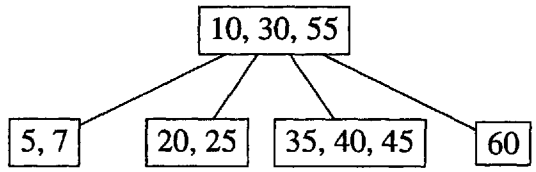
14. [5 marks] My friend Charles needs to store a lot of URLs in a hash table. He has designed a hash function "*chash*" for this. Unfortunately, he has found that *chash* has a high rate of collisions.

    He tries to improve the situation by adding universal hashing. Using the universal family from this course, he picks a large enough prime p and an array size m. His program picks random a and b with $0 < a < p$ and $0 \le b < p$ at startup, then the new hash function is

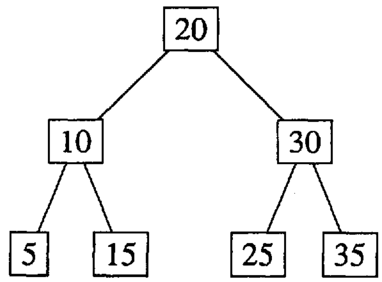
    $$newhash(url) = h_{a,b}(chash(url)) = ((a \cdot chash(url) + b) \bmod p) \bmod m$$

    Explain why *newhash* has a high rate of collisions too.

15. [5 marks] Insert 50 into this (2,4)-tree. Draw the final tree.



16. [5 marks] Delete 35 from this (2,4)-tree. Draw the final tree.



17. [10 marks] Many computer puzzle games contain riddles based on anagrams. An English word is an anagram of another English word iff one is a permutation of the other.

    Examples: "low" is an anagram of "owl", but not an anagram of "wool". "polo" is an anagram of "loop".

    I want a computer program that reads a file of all English words at startup just once, stores them in some data structure, then repeatedly takes an input string and finds all anagrams of the input string. This will help me solve those riddles. For simplicity, assume lowercase letters.

    Describe an efficient data structure for storing the English words and an efficient algorithm for looking up all anagrams of an input string.

18. [10 marks] Here is a game: There is an undirected graph. There are two tokens, T_1 and T_2; initially, T_1 is at vertex v_1, and T_2 is at vertex v_2. The goal is to move T_1 to v_2 and to move T_2 to v_1, i.e., exchange their places. The rules are:

- At each step, you may move only one token, and only move it to a vertex adjacent to its current position.
- The two tokens can never be at the same vertex at the same time.
- Use the fewest number of steps possible.

Describe an efficient algorithm to compute a solution or determine that there is no solution.

End of questions. Total pages = 16. Total marks = 120.