Instructor: Dr. Sartaj Sahni
Summer, 2004

Advanced Data Structures
(NTU AD 711R)
Exam 1

CLOSED BOOK
60 Minutes

Name: _____

NOTE:
1. **For all problems, use only the algorithms discussed in class/text.**
2. All answers will be graded on correctness, efficiency, clarity, elegance and other normal criteria that determine quality.
3. The points assigned to each question are provided in parentheses.

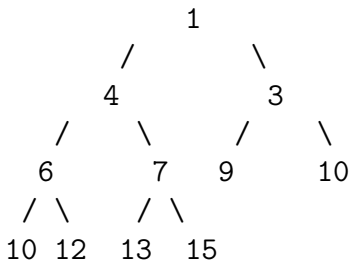
1. (10) Suppose that we use a stack data type to implement a *First-In-First-Out(FIFO) queue*. Assume that a stack has *push*() and *pop*() operations and the actual cost of each operation is 1.
   Specifically, we implement a queue using two stacks $Stack_1$ and $Stack_2$ and two queue operations *enqueue*() and *dequeue*() as follows:

   - *enqueue*(x): push x onto $Stack_1$
   - *dequeue*(): If $Stack_2$ is not empty, then simply *pop* from $Stack_2$ and return the element. If $Stack_2$ is empty, *pop* all the elements of $Stack_1$, *push* them into $Stack_2$, then *pop* from $Stack_2$ and return the result.

   What is the smallest integer amortized cost of each *enqueue*() and *dequeue*() operation? (Note: Do not consider the cost of checking whether $Stack_2$ is empty or not. Measure the cost only in terms of the number of *push*() and *pop*() operations. The amortized cost of *enqueue*() may be different from that of *dequeue*().)

2. (8) Construct the binary $Huffman$ tree for the six weights a=45, b=13, c=12, d=16, e=9, and f=5. What is the weighted external path length of your tree?

3. (10) Consider the priority queue represented by the given min leftist tree. Show the modified tree under each of the following operations. (Note: The two operations (a) and (b) are independent. Each of them starts from the given tree.)



(a) (5) *Insert* 8. Label each node of the resulting tree with its s-value($shortest(x)$).

(b) (5) Perform a $RemoveMin()$ operation on the given tree above. Show each step.

4. (12) Perform the following operation sequence on an initially empty *min binomial heap* (showing each step).

   Insert(5), Insert(9), Insert(11), Insert(3),Insert(7),Insert(25), RemoveMin, RemoveMin, Insert(4), Insert(2),Insert(6),RemoveMin.

   (Note: For consistency in solution, if you have three binomial trees of the same degree in the intermediate steps, please leave the binomial tree with the largest root, and combine the other binomial trees.)