Assignment 1

Due Date: Before the class on August 22.

Some Important Points

1. Make a group of at most 2 people. The group members remain same till the end of this course.
2. Collaboration is encouraged. However, each group should write its final answer separately.
3. Please mention names of all the people with whom you have discussed the question.

1. (5 points) Implement a queue using two stacks. Analyze the running time of ENQUEUE(a) and DEQUEUE().
2. (5 points) Implement a stack using two queues. Analyze the running time of PUSH(a) and POP().
3. (10 points) Two linked list $\mathcal{L}_1$ and $\mathcal{L}_2$ merge at one particular node after which both share all the subsequent nodes (See Figure 1). If l_1 and l_2 are the size of list $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively, design an algorithm that finds the nodes at which the two list merge in $O(l_1 + l_2)$ running time. Your algorithm should take only $O(1)$ extra space (over the space required to store the two list) and initially you only have access to head of both the list.
4. (10 points) Given k sorted arrays(ascending order) $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$ such that $\sum_{i=1}^{k}$ (size of $\mathcal{A}_i$) = n, design an algorithm that can merge these k arrays to a single sorted array in $O(n \log k)$ running time.

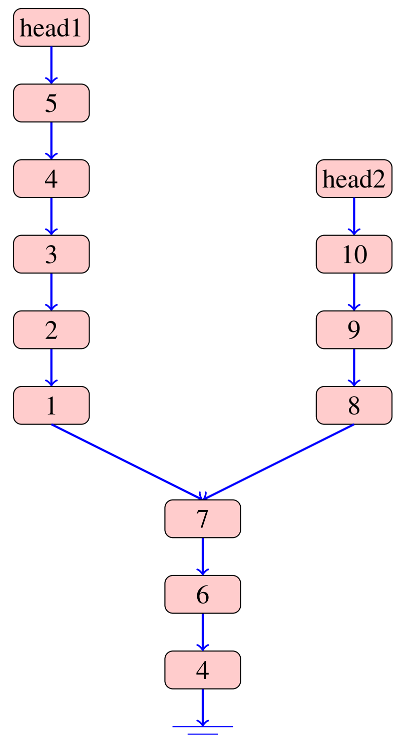

Figure 1: $\mathcal{L}_1$ and $\mathcal{L}_2$ merge at node with value 7. l_1 = 8 and l_2 = 6.

5. (10 points) In the class, we saw the following two procedures to insert an element from the in the heap and delete the minimum element from the heap.

Make a new node v with $v.value \leftarrow a$, $v.left \leftarrow null$ and $v.right \leftarrow null$;
Find the right place for v on the last level ;
SHIFT-UP(v);

Figure 2: INSERT(a)

Let v be the last node at the last level of the heap;
$root.value \leftarrow v.value$;
Deallocate the memory of v;
SHIFT-DOWN($root$);

Figure 3: DELETE-MIN()

Except the red step in the above procedure, (in the class) we showed that both INSERT(a) and DELETE-MIN() takes $O(\log n)$ time. Design a procedure which can execute the red step in $O(1)$ time.